

# Tru64 UNIX

---

## AdvFS Administration

Part Number: AA-RH96B-TE

**August 2000**

**Product Version:** Tru64 UNIX Version 5.1 or higher

This guide provides task-based information and instruction for understanding, configuring, troubleshooting, and tuning the AdvFS file system. It is designed primarily for system administrators.

---

© 1995, 1996, 1997, 1998, 1999, 2000 Compaq Computer Corporation

COMPAQ, the Compaq logo, and the Digital logo are registered in the U. S. Patent and Trademark Office. TruCluster and Tru64 are trademarks of Compaq Computer Corporation.

UNIX and Motif are registered trademarks and The Open Group is a trademark of The Open Group in the United States and other countries. Other product names mentioned herein may be the trademarks of their respective companies.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Compaq Computer Corporation or an authorized sublicensor.

Compaq Computer Corporation shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

---

# Contents

## Preface

## 1 Configuring the File System

1.1	File System Design .....	1-1
1.1.1	Filesets, Domains and Volumes .....	1-2
1.1.2	Transaction Log .....	1-3
1.1.3	File Storage Allocation .....	1-3
1.2	Setting Up AdvFS .....	1-5
1.3	Volumes .....	1-6
1.3.1	Volume Structure .....	1-6
1.3.2	Volume Attributes .....	1-7
1.4	Domains .....	1-7
1.4.1	Configuring Domains .....	1-7
1.4.2	The /etc/fdmns Directory .....	1-8
1.4.3	Version 5 Domains .....	1-9
1.4.3.1	Creating a Domain with DVN4 .....	1-9
1.4.3.2	Upgrading a Domain to DVN4 .....	1-10
1.4.4	Displaying Domain Information .....	1-11
1.4.5	Creating a Domain .....	1-11
1.4.6	Increasing the Size of a Domain .....	1-13
1.4.7	Reducing the Size of a Domain .....	1-15
1.4.8	Removing a Domain .....	1-17
1.4.9	Renaming a Domain .....	1-17
1.5	Filesets .....	1-18
1.5.1	Configuring Filesets .....	1-18
1.5.2	Designating Filesets in the /etc/fstab File .....	1-19
1.5.3	Displaying Fileset Information .....	1-20
1.5.4	Creating a Fileset .....	1-21
1.5.5	Upgrading a Fileset .....	1-21
1.5.6	Mounting a Fileset .....	1-22
1.5.7	Unmounting a Fileset .....	1-22
1.5.8	Removing a Fileset .....	1-22
1.5.9	Renaming a Fileset .....	1-23
1.5.10	Clone Filesets .....	1-24
1.5.10.1	Creating a Clone Fileset .....	1-25

1.5.10.2	Mounting and Unmounting a Clone .....	1-26
1.5.10.3	Removing a Clone .....	1-26
1.5.10.4	Renaming a Clone .....	1-26
1.6	Configuring an AdvFS root File System .....	1-26
1.6.1	Mounting the root File System in Single-User Mode .....	1-27
1.6.2	Changing the Name of the root Domain .....	1-27
1.6.3	Changing the Name of the root Fileset .....	1-27
1.7	Using AdvFS with LSM .....	1-29
1.7.1	Setting Up AdvFS with LSM Volumes .....	1-30
1.7.2	Mirroring LSM Volumes Under AdvFS .....	1-31
1.7.3	Striping LSM Volumes and Striping AdvFS Files .....	1-32

## 2 Managing Quotas

2.1	Introducing Quotas .....	2-1
2.1.1	Quota Limits .....	2-1
2.1.2	Grace Period .....	2-2
2.2	User and Group Quotas .....	2-3
2.2.1	Quota Files .....	2-3
2.2.2	Setting Quotas and Grace Periods .....	2-4
2.2.2.1	User Quotas Example .....	2-5
2.2.2.2	User Grace Period Example .....	2-6
2.2.2.3	Group Quotas Example .....	2-7
2.2.2.4	Group Grace Period Example .....	2-8
2.2.3	Setting Quotas for Multiple Users and Groups .....	2-9
2.2.3.1	Prototype User Example .....	2-9
2.2.3.2	Prototype Group Example .....	2-10
2.2.4	Verifying File and Disk Space Usage .....	2-10
2.2.4.1	Printing the Tag and Full Path Name for Each File ...	2-11
2.2.4.2	Summarizing Fileset Ownership .....	2-11
2.2.4.3	Displaying Disk Usage and Limits .....	2-12
2.2.4.4	Verifying Quota Consistency .....	2-13
2.2.4.5	Summarizing Quotas by Fileset .....	2-13
2.2.5	Activating Quotas at System Start-up .....	2-15
2.2.6	Activating Quotas Manually .....	2-15
2.2.7	Deactivating Quotas .....	2-16
2.3	Fileset Quotas .....	2-16
2.3.1	Quota Files .....	2-17
2.3.2	Setting Quotas and Grace Periods .....	2-17
2.3.2.1	Fileset Quotas Example .....	2-17
2.3.2.2	Setting the Grace Period .....	2-18
2.3.3	Setting Quotas for Multiple Filesets .....	2-19

2.3.4	Verifying File and Disk Space Usage .....	2-20
2.3.4.1	Displaying Used and Available Disk Space for Filesets .....	2-20
2.3.4.2	Displaying Used and Available Disk Space for Domains and Filesets .....	2-21
2.3.4.3	Displaying Fileset Attributes .....	2-22
2.3.4.4	Displaying Domain Attributes .....	2-22
2.3.5	Activating Quotas .....	2-23
2.3.6	Deactivating Quotas .....	2-23

### 3 Backing Up and Restoring

3.1	Backing Up Data .....	3-2
3.1.1	Unique Features of the vdump Command .....	3-2
3.1.2	Dumping to Tape .....	3-3
3.1.3	Dumping Subdirectories .....	3-3
3.1.4	Dumping to Filesets .....	3-3
3.1.5	Dumping to Block 0 .....	3-3
3.1.6	Compressing Savesets .....	3-4
3.1.7	Dumping with Error Protection .....	3-4
3.1.8	Backup Level .....	3-5
3.1.9	Listing Saved Files .....	3-5
3.1.10	Dumping and Restoring Files Remotely .....	3-5
3.2	Cloning for Online Backup .....	3-6
3.3	Cloning to Back Up Databases .....	3-6
3.4	Restoring Data .....	3-7
3.4.1	Unique Features of the vrestore Command .....	3-7
3.4.2	Restoring Files .....	3-8
3.4.3	Restoring Quotas .....	3-8
3.4.4	Restoring Selected Savesets .....	3-8
3.5	AdvFS and NetWorker .....	3-9

### 4 Optimizing Performance

4.1	AdvFS Organization .....	4-1
4.1.1	Configuring Domains and Filesets .....	4-1
4.1.2	Configuring Volumes .....	4-1
4.1.3	Improving Transaction Log Performance .....	4-2
4.1.4	Improving Data Consistency .....	4-4
4.1.5	Improving Data Transfer Rate with Direct I/O .....	4-6
4.2	Monitoring Performance .....	4-7
4.3	Tuning AdvFS .....	4-7

4.3.1	Defragmenting a Domain .....	4-7
4.3.1.1	Choosing to Defragment .....	4-9
4.3.1.2	Defragment Example .....	4-10
4.3.2	Balancing a Multivolume Domain .....	4-12
4.3.2.1	Choosing to Balance .....	4-13
4.3.2.2	Balance Example .....	4-13
4.3.3	Moving Files to Different Volumes .....	4-14
4.3.3.1	Choosing to Migrate .....	4-16
4.3.3.2	Migrate Example .....	4-16
4.3.4	Striping Files .....	4-17
4.3.4.1	Choosing to Stripe an AdvFS File .....	4-18
4.3.4.2	AdvFS Stripe Example .....	4-18
4.3.4.3	Removing AdvFS Striping .....	4-19
4.3.5	Data Cache Tuning .....	4-20
4.3.6	Changing Attributes to Improve System Performance .....	4-20
4.3.7	Controlling Domain Panic Information .....	4-21
4.4	Using a Trashcan .....	4-21

## 5 Troubleshooting

5.1	Disk File Structure Incompatibility .....	5-1
5.1.1	Utility Incompatibility .....	5-1
5.1.2	Avoiding Metadata Incompatibility .....	5-2
5.2	Memory Mapping, Direct I/O and Data Logging Incompatibility .....	5-2
5.3	Handling Poor Performance .....	5-2
5.4	Handling Disk Problems .....	5-4
5.4.1	Checking Free Space and Disk Usage .....	5-4
5.4.2	Reusing AdvFS Volumes .....	5-5
5.4.3	Dumping to Block 0 .....	5-6
5.4.4	Disk Space Usage Limits .....	5-6
5.4.5	Verifying File System Consistency .....	5-6
5.4.6	Salvaging File Data from a Damaged Domain .....	5-8
5.4.7	"Can't Clear a Bit Twice" Error Message .....	5-11
5.4.8	Recovering from a Domain Panic .....	5-12
5.4.9	Recovering from Filesets Mounted Read-Only .....	5-14
5.5	Restoring an AdvFS File System .....	5-15
5.5.1	Restoring the /etc/fdmns Directory .....	5-15
5.5.1.1	Reconstructing the /etc/fdmns Directory Manually .....	5-16
5.5.1.2	Reconstructing the /etc/fdmns Directory Using advscan .....	5-17
5.5.2	Recovering from Volume Failure .....	5-20
5.5.3	Recovering from Failure of the root Domain .....	5-22

5.5.4	Restoring a Multivolume usr Domain .....	5-23
5.6	Recovering from a System Crash .....	5-25
5.6.1	Saving Copies of System Metadata .....	5-25
5.6.2	Physically Moving an AdvFS Disk .....	5-26
5.6.3	Log File Inconsistency .....	5-27

## 6 Managing the Advanced File System with the AdvFS GUI

6.1	Installing the GUI .....	6-1
6.2	Components of the GUI .....	6-2
6.2.1	GUI Agent (advfsd) .....	6-2
6.2.2	GUI (dtadvfs) .....	6-3
6.2.2.1	GUI Security .....	6-3
6.2.2.2	Ignoring Disks .....	6-4
6.2.2.3	Log Files .....	6-4
6.2.2.4	Help .....	6-4
6.2.2.5	Monitoring Remote File Systems .....	6-5
6.2.2.6	Managing LSM Volumes .....	6-5
6.3	Displaying File System Information .....	6-6
6.3.1	Choosing Units for Displaying Information .....	6-7
6.3.2	Main Window .....	6-7
6.3.3	Devices and Volumes Window .....	6-8
6.4	Performing File System Operations .....	6-9
6.4.1	Operations on a Domain .....	6-9
6.4.1.1	Creating a Domain .....	6-9
6.4.1.2	Setting a Free Space Alert for a Domain .....	6-11
6.4.1.3	Enlarging a Domain .....	6-11
6.4.1.4	Removing a Domain .....	6-12
6.4.1.5	Renaming a Domain .....	6-12
6.4.2	Operations on a Fileset .....	6-13
6.4.2.1	Creating a Fileset .....	6-13
6.4.2.2	Mounting a Fileset .....	6-14
6.4.2.3	Unmounting a Fileset .....	6-14
6.4.2.4	Setting a Free Space Alert for a Fileset .....	6-14
6.4.2.5	Setting Fileset Quotas .....	6-15
6.4.2.6	Removing a Fileset .....	6-15
6.4.2.7	Renaming a Fileset .....	6-15
6.4.3	Operations on a Clone Fileset .....	6-16
6.4.3.1	Creating a Clone Fileset .....	6-16
6.4.3.2	Mounting a Clone .....	6-16
6.4.3.3	Unmounting a Clone .....	6-17
6.4.3.4	Removing a Clone .....	6-17

6.4.4	Operations on a Volume .....	6-17
6.4.4.1	Setting a Free Space Alert for a Volume .....	6-18
6.4.4.2	Adding a Volume .....	6-18
6.4.4.3	Removing a Volume .....	6-18
6.4.5	Maintaining AdvFS .....	6-18
6.4.5.1	Balancing a Multivolume Domain .....	6-19
6.4.5.2	Defragmenting a Domain .....	6-19
6.5	Troubleshooting GUI Operation .....	6-19

## **A Accessing File System Information with the SysMan Menu**

### **B AdvFS Commands**

B.1	AdvFS Base System Commands .....	B-1
B.2	AdvFS Utilities Commands .....	B-3
B.3	Comparison of AdvFS and UFS Commands .....	B-4

### **C Converting File Systems**

C.1	Converting a /usr File System to AdvFS .....	C-1
C.1.1	Using a Backup Tape .....	C-1
C.1.2	Using an Intermediate File .....	C-3
C.1.3	Using a Second Disk .....	C-4
C.2	Converting the root File System to AdvFS .....	C-6
C.3	Converting a Data File System to AdvFS .....	C-7
C.3.1	Using a Backup Tape .....	C-7
C.3.2	Using an Intermediate File .....	C-9
C.3.3	Using a Second Disk .....	C-10
C.3.4	Using a Second System .....	C-11
C.4	Converting from AdvFS to UFS .....	C-13
C.4.1	Converting the root File System to UFS .....	C-13
C.4.2	Converting a Fileset to UFS .....	C-14

## **Glossary**

## **Index**

## **Figures**

1-1	AdvFS File System Design .....	1-2
1-2	Single-Volume Domain .....	1-12
1-3	Enlarging a Domain .....	1-13



1-4	Cloning a Fileset .....	1-24
4-1	Defragmenting a Domain .....	4-8
4-2	Balancing a Domain .....	4-12
4-3	Migrating Files .....	4-15
6-1	Main Window Showing Disk Usage Information .....	6-8
6-2	Main Window Showing Fileset Quota Information .....	6-8
6-3	Devices and Volumes Window .....	6-9

## Tables

2-1	Quota Size Limits .....	2-2
2-2	User and Group Usage Commands .....	2-11
2-3	Fileset Disk Usage Commands .....	2-20
4-1	Trashcan Commands .....	4-22
5-1	Disk Space Usage Information Commands .....	5-4
5-2	Fileset Anomalies and Corrections .....	5-18
5-3	Fileset Anomalies and Corrections .....	5-18
5-4	Fileset Anomalies and Corrections .....	5-19
6-1	AdvFS GUI Subsets .....	6-1
6-2	Disk Space Requirements for the AdvFS GUI .....	6-2
6-3	GUI Troubleshooting .....	6-20
B-1	AdvFS Configuration Commands .....	B-1
B-2	AdvFS Information Display Commands .....	B-1
B-3	AdvFS Backup Commands .....	B-2
B-4	AdvFS Check and Repair Commands .....	B-2
B-5	AdvFS Quota Commands .....	B-2
B-6	AdvFS Disk Structure Commands .....	B-3
B-7	AdvFS Utilities Commands .....	B-3
B-8	Comparison of AdvFS and UFS Commands .....	B-4



---

## Preface

Advanced File System (AdvFS) is the default file system for the Compaq Tru64 UNIX operating system. AdvFS is a log-based file system that provides flexibility, compatibility, data availability, and high performance. AdvFS takes advantage of the 64-bit computing environment and is designed to handle files and filesets of nearly 16 terabytes. For hardware and software requirements, see the Tru64 UNIX Software Product Description and the AdvFS Utilities Software Product Description.

AdvFS Utilities, which is licensed separately from the Tru64 UNIX operating system, provides additional file management capabilities and a graphical user interface (GUI) to simplify system administration. The AdvFS GUI, which runs under the Common Desktop Environment (CDE), features menus, graphical displays, and comprehensive online help that make it easy to perform AdvFS operations. In addition, the GUI displays summarized system status information. It includes utilities to add volumes, create clones, stripe files, and balance domains.

This guide describes AdvFS and AdvFS Utilities in detail. It provides information on features and functions, and it gives suggestions on how to use these functions. Many functions can be accomplished through either the command line, the GUI, or the SysMan Menu (see `sysman(8)`).

## Introducing AdvFS

The configuration of AdvFS differs from the traditional UNIX file system. In AdvFS, the physical storage layer is managed independently of the directory layer. System administrators can add and remove storage without unmounting the file system or halting the operating system. As a result, configuration planning is less complicated and more flexible.

From a user's perspective, AdvFS behaves like any other UNIX file system. End users can use the `mkdir` command to create new directories, the `cd` command to change directories, and the `ls` command to list directory contents. AdvFS logical structures, quota controls, and backup capabilities are based on traditional file system design. AdvFS replaces or eliminates several standard commands, such as `newfs`, `dump`, `restore`, and `fsck`. AdvFS commands and utilities and a comparison of AdvFS and UFS commands are detailed in Appendix B.

Without taking an AdvFS system off line, system administrators can perform backups, file system reconfiguration, and file system tuning. End

users can retrieve their own unintentionally deleted files from predefined trashcan directories or from clone filesets without assistance from system administrators.

AdvFS supports multivolume file systems, which enables file-level striping (spreading data to more than one volume) to improve file transfer rates for I/O-intensive applications. Logical Storage Manager (LSM), which allows volume-level striping, can be incorporated into AdvFS configurations.

## Advanced File System Features

The following table lists the main features and benefits of AdvFS.

### AdvFS Features and Benefits

Feature	Benefit
Rapid crash recovery	Write-ahead logging eliminates the need to use the <code>fsck</code> utility to recover from a system failure. The file system recovery time is rapid and is independent of file system size.
Extended capacity, scalability	The design supports large-scale storage systems.
High performance	An extent-based file allocation scheme consolidates data transfer.
Disk spanning*	A file or file system can span multiple disks within a shared storage pool.
Unified buffer cache (UBC)	This cache interacts with the virtual memory system to dynamically adjust the amount of physical memory being used to cache file data.
Online defragmentation	System performance improves by defragmenting the data on the disk while the system remains in use. Defragmentation makes file data more contiguous on the storage medium.
Online resizing*	The size of the file system can be dynamically changed by adding or removing disk volumes while the system remains in use. This enables both online storage configuration and online file system maintenance.
File-level striping*	Distributing file data across multiple disk volumes improves file transfer rates.
Online backup*	File system contents can be backed up to media without interrupting the work flow of system users by using fileset clones.

### **AdvFS Features and Benefits (cont.)**

<b>Feature</b>	<b>Benefit</b>
File undelete*	Users can recover deleted files without assistance from system administrators.
Quotas	AdvFS supports quotas for users and groups as well as for filesets.
Graphical user interface*	The GUI simplifies file system management by organizing AdvFS functions into menu-selected tasks and by displaying file system status.

\* This feature requires the optional AdvFS Utilities license.

## **What's New in Tru64 UNIX Version 5.1**

The following features have been added to AdvFS:

- AdvFS cache tuning is done through the UBC parameters (see Section 4.3.5).
- Improved disk structure that increases the number of files the domain can track (see Section 1.4.3)
- A disk salvage utility that can recover information at the block level from disks that have been damaged (see Section 5.4.6)
- An improved directory structure that increases the speed of file creation and access (see Section 1.4.3.2)
- Enhanced `vdump` and `vrestore` command capability (see Chapter 3)
- Remote device support for backup and restore (see Chapter 3)
- Increased quota limits (see Chapter 2)
- Direct I/O to allow unbuffered, synchronous I/O (see Section 4.1.5)
- Smooth sync option to promote continuous I/O (see Section 4.3.6)
- New utilities (such as `vdff`, which displays disk usage for domains and filesets) (see Section 2.3.4.2)
- Metadata display utilities (see Section 5.6.1)

## Conventions

This guide uses the following conventions:

<code>%</code>	A percent sign represents the C shell system prompt.
<code>\$</code>	A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
<code>#</code>	A number sign represents the superuser prompt.
<code>% cat</code>	Boldface type in interactive examples indicates typed user input.
<i>file</i>	Italic (slanted) type indicates variable values, placeholders, and function argument names.
<code>[ ] {   }</code>	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
<code>...</code>	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
<code>cat(1)</code>	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, <code>cat(1)</code> indicates that you can find information on the <code>cat</code> command in Section 1 of the reference pages.
<code>Ctrl/x</code>	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, <span style="border: 1px solid black; padding: 2px;">Ctrl/C</span> ).

## License Registration

AdvFS is the file system default on the operating system. AdvFS Utilities is a separately licensed product that provides additional processing capabilities. Before you can use the file system utilities, you must register a license product authorization key (PAK) for AdvFS Utilities. Contact your software support organization for additional information.

## Related Information

Other useful documentation includes:

- System Administration
- Installation Guide
- Logical Storage Manager
- Programmer's Guide
- Technical Overview
- Software License Management
- System Configuration and Tuning
- Software Product Descriptions for Tru64 UNIX and AdvFS Utilities
- Cluster Administration





---

## Configuring the File System

AdvFS differs from the traditional UNIX File System (UFS). The UFS model is rigid. Each disk (or disk partition) contains one separate file system; you mount the file system into the logical name space using mount points.

The directory hierarchy layer of UFS is bound tightly to the physical storage layer. When a file system becomes full, this tight binding makes it impossible to move selected files onto another disk without changing the full path names of those files. The task of dividing a logical directory into directory subtrees and mapping the subtrees onto separate disks requires careful consideration. Even with extensive planning, adjustments to the directory structure are limited with the UFS model.

In contrast, with AdvFS you can modify your system configuration at any time without taking down the system. As your system requirements change, AdvFS allows you to easily adjust your storage up or down to meet your requirements.

### 1.1 File System Design

AdvFS consists of two distinct layers: the directory hierarchy layer and the physical storage layer. The directory hierarchy layer implements the file-naming scheme and POSIX-compliant functions such as creating and opening files, or reading and writing to files. The physical storage layer implements write-ahead logging, caching, file allocation, and physical disk I/O functions.

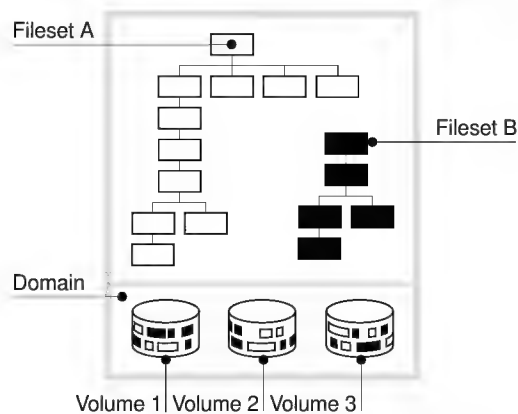
The decoupled file system structure enables you to manage the physical storage layer apart from the directory hierarchy layer. This means that you can move files between a defined group of disk volumes without changing path names for your files. Because the path names remain the same, the action is completely transparent to end users.

AdvFS can incorporate Logical Storage Manager (LSM) volumes into the file system structure. AdvFS configured with LSM improves file system reliability and availability because AdvFS can take advantage of LSM features (see Section 1.7).

### 1.1.1 Filesets, Domains and Volumes

AdvFS implements two unique file system concepts: filesets and domains. Filesets and domains enable a two-layer file system structure in which the physical storage layer of the file system is independent of its directory hierarchy. With this unique architecture, you can manage the logical structure of your file systems independently of the storage volumes that contain them. Figure 1-1 is a representation of this structure.

**Figure 1-1: AdvFS File System Design**



A fileset follows the logical structure of a traditional UNIX file system. It is a hierarchy of directory names and file names, and it is what you mount. AdvFS goes beyond the traditional file system by allowing you to create multiple filesets that share a common pool of storage called a domain. See Section 1.5 for more information about filesets.

A domain is managed separately from the directory structure. You can add or remove volumes without affecting the directory structure. See Section 1.4 for more information.

A volume is any mechanism that behaves like a UNIX block device: an entire disk, a disk partition, or an aggregate volume provided by a logical storage manager (see Section 1.3). When first created, all domains consist of a single volume. If you have the optional AdvFS Utilities (see License Registration), you can transform a single-volume domain into a multivolume domain by adding one or more volumes.

Domain names must be unique. You cannot use slash (/), pound (#), colon (:), asterisk (\*), question mark (?), tab, newline, formfeed, return, vertical tab, and space characters. Filesets that are not in the same domain can have the same name and file names must be unique in the directory.

### 1.1.2 Transaction Log

AdvFS is a log-based file system that employs write-ahead logging to ensure the integrity of the file system. Write-ahead logging means that modifications to the metadata (the on-disk file system structures) are completely written to a transaction log before the actual changes are written to disk. The log is implemented as a circular file buffer so that the contents of the transaction log are written to disk at regular intervals. By committing only completed transactions to disk, the file system is not left in an inconsistent state after an unexpected system crash.

When a domain is created, AdvFS creates a transaction log for it. The first time the domain is activated, 4 megabytes of storage are allocated for the log. It is separate from the user data but can be stored on the same device.

During crash recovery, AdvFS reads the transaction log to confirm file system transactions. All completed transactions are committed to disk and uncompleted transactions are undone. The number of uncommitted records in the log, not the amount of data in the file system, dictates the speed of recovery. This means that recovery usually takes only a few seconds. Traditional UNIX file systems rely on the `fsck` utility to recover from a system failure. The `fsck` utility can take hours to check and repair a large file system.

By default, only file system structures are logged, but you can choose to log file data and change the way your system writes to storage (see Section 4.1.4). A file that has data logging turned on will remain internally consistent in the event of a system crash. However, enabling data logging can slow system performance.

### 1.1.3 File Storage Allocation

Files are not static; their space requirements change over time. To maintain contiguous file placement without overallocating space on the disk, AdvFS uses a unique file storage allocation scheme.

Key features of storage allocation are:

- Extents

An extent is a contiguous area of disk space that AdvFS allocates to a file. Extents are composed of sets of one or more 8-kilobyte pages. When storage is added to a file, it is grouped in extents. There is an extent map for each volume on which the file system resides.

File I/O is most efficient when there are few extents. If a file consists of many small extents, it takes more I/O processing to read or write that file.

Given the dynamic nature of a file system, the file-storage allocation cannot always guarantee contiguous placement of pages. The following factors affect placement:

- Excessive disk fragmentation

When a disk is fragmented, there are many small free spaces, so AdvFS writes data to isolated physical pages, based on availability, instead of writing to contiguous pages. This may result in files with many extents.

- Multiple users

When there are many users on a system, requests for space increase, decreasing the likelihood of contiguous file allocation.

If you have a domain with many extents, you can decrease the number by running the `defragment` utility (see Section 4.3.1).

- Preallocation

Each time a file is appended, AdvFS adds pages to the file by preallocating one-fourth of the file size, up to 16 pages. Excess preallocated space is truncated when the file is closed.

For multivolume domains, new files are allocated sequentially across volumes. Volumes that are more than 86% full (allocated) are not used for new file allocation unless all volumes are more than 86% full. When data is appended to existing files, storage is allocated on the volume on which the file was initially allocated until the volume is full.

- Fragments

AdvFS writes files to disk in sets of 8-kilobyte pages. In files where holding the last bytes in an 8-kilobyte page would represent a waste of more than 5% of the allocated space, a file fragment is created. The fragment, which ranges in size from 1 kilobyte to 7 kilobytes, is allocated from the `frag` file, a special file used for storing the file fragments. This technique considerably reduces the amount of unused, wasted disk space.

- Sparse files

A sparse file is one that does not have disk storage reserved for some of its pages. Reading a sparse file at a page that does not have storage reserved returns zeroes. Writing to a page that does not have storage reserved allocates a page of disk storage.

A sparse file is created by setting a file's end-of-file with `ftruncate(2)` or by using `lseek(2)` and `write(2)` to skip over unwritten sections of the file.

Core files are sparse files. They have large areas with no information and don't use up disk blocks for locations where no data exists. Quota files are sparse because they are indexed by user ID. If there are gaps in the user IDs, there may be sections of the file with no data.

In contrast, database files generally reserve storage for the entire file even when data is not available. Database applications generally write zeroes in the pages that have no useful data so that storage is allocated. In addition, by writing the data sequentially, a database file is created with large sets of contiguous pages and a small number of extents.

To examine the length of a sparse file, including the pages that don't have disk storage, use the `ls` command with the `-l` option. The `ls` command with the `-s` option shows the amount of storage actually used by the file. The `du` command gives the same information as the `ls -s` command.

## 1.2 Setting Up AdvFS

When planning your configuration, consider setting up the `root` and `/usr` file systems on AdvFS. Using AdvFS as the `root` file system enables booting from an AdvFS domain. By having the `/usr` file system on AdvFS, you can significantly reduce the amount of time your system is down after a system failure.

You can set up AdvFS to resemble a traditional UFS configuration with one partition (volume) per domain and each domain containing one fileset. If you have the optional AdvFS Utilities, when space is needed you can add volumes (except to the local `root` which is restricted to one volume) to increase the size of existing domains. You do not have to change anything in the existing configuration.

The minimum configuration needed for an active AdvFS file system is one domain and one mounted fileset.

To create an active domain:

1. Create a domain (see Section 1.4.5).
2. Create a fileset (see Section 1.5.4).
3. Create a mount-point directory (see Section 1.5.6).
4. Mount a fileset (see Section 1.5.6).

You can start with one fileset per domain or you can have many. See Section 1.4.1 and Section 1.5.1 for guidelines.

You can automatically mount a fileset at system reboot (see Section 1.5.2) or use the AdvFS GUI or the SysMan Menu to perform this activity. See Appendix B for a complete list of AdvFS commands.

For more detailed information on AdvFS configuration, see System Configuration and Tuning.

## 1.3 Volumes

A volume on AdvFS can be a single disk partition, an entire disk, an aggregate volume provided by Logical Storage Manager (LSM), or hardware or software redundant array of independent disks (RAID) storage.

A volume can be assigned to only one domain. It is associated with its domain by a domain ID, which is automatically stored in the domain attributes table of the volume. Each volume in a domain is assigned a volume index number, starting with 1, when it is initialized. Numbers are reused when volumes are removed and new ones added. When a volume is removed from a domain, the domain ID is cleared in the domain attributes table.

If you have the optional AdvFS Utilities, adding volumes allows you to create a multivolume domain, increase the disk capacity of an existing domain, increase the storage available for the filesets, and perform preventative disk maintenance. You can add volumes immediately after creating the domain, even before creating and mounting filesets. To perform preventative disk maintenance, you can add a new volume to the domain, migrate your files to the new volume, and then remove the old volume.

For AdvFS to function properly, the number of volumes in a domain with the same domain ID must remain consistent with the number of volumes. In addition, each domain is defined by an entry in the `/etc/fdmns` directory (see Section 1.4.2). This directory must be up-to-date; that is, the domain entries must correctly reference the volumes associated with the domains. The number of links to the volumes in the `/etc/fdmns` directory must equal the number of volumes.

All volumes are labeled (see `disklabel(8)`) either `unused` or with the file system for which they were last used (for example, `advfs`). You can only add a volume that is labeled `unused` to a domain (see Section 5.4.2).

### 1.3.1 Volume Structure

Each volume in an AdvFS domain contains the following structures:

- A bitfile metadata table (BMT), which is used to store file data structure (metadata), including file attributes, file extent maps, fileset attributes, and the POSIX file statistics.
- A storage bitmap, which is used to track free and allocated disk space.
- A miscellaneous metadata bitfile, which maps areas of the volume that do not represent AdvFS metadata, such as the disk label and boot blocks.

In addition to these structures, each domain has the following structures on one volume in the domain:

- A transaction log, which stores all metadata changes until they are written to disk.
- A root tag directory, which defines the location of all filesets in the domain.

This information is provided only to show how the volume is structured for AdvFS. You cannot change the way AdvFS configures the volume.

### 1.3.2 Volume Attributes

AdvFS volumes are configured with attributes that determine how data is read, cached, written, and consolidated. When an AdvFS volume is incorporated into a domain, either by creating the initial domain or by adding a volume, the default volume attributes are set. Modifying these default attributes may improve performance in some system configurations.

To display or modify the current volume attributes, use the SysMan “Manage an AdvFS Domain” or, from the command line, enter the `chvol` command:

```
chvol device_name file_domain
```

See Chapter 4 and System Configuration and Tuning for more complete information on modifying attributes to improve system performance.

## 1.4 Domains

A domain is the physical storage layer of the AdvFS file system. It is a defined pool of physical storage that can contain one or more volumes. Because this storage is managed separately from the directory structure (see Section 1.4.2), you can expand and contract the size of the domain by adding or removing volumes. You can move files between disks in the domain without changing file path names.

Each domain has a unique ID that is an encoding of the date and time the domain was created. If you change the name of the domain, it does not affect this ID.

### 1.4.1 Configuring Domains

How you configure domains depends on your organization’s needs. You can assign all available storage to a few domain or group specific partitions or disks into many domains. You must also take into account how you will configure your filesets (see Section 1.5.1). With AdvFS Utilities, you can assign multiple volumes to a domain and distribute the filesets and files across the volumes.

Establishing multiple domains allows greater control over your physical resources. You can create domains to be used by specific projects, groups of

users, departments, or any division that makes sense for your organization. For example, you could create domains for each of your organization's departments, such as engineering, finance, and personnel.

There are a number of factors to consider when configuring domains:

- To maintain high performance, avoid splitting a disk between two domains. For example, do not add partition `g` to one domain and partition `h` of the same disk to another domain.
- Adding one partition containing the entire disk (typically, partition `c`) to a domain is preferable to adding several partitions on the same disk. Doing so may degrade performance. You will get an error message if you try to assign overlapping partitions to a domain (see Section 1.4.6).
- It is generally more efficient to spread your domain over several disks, assuming the disks are the same type and speed. For example, a domain with three volumes on separate disks is more efficient than one with three partitions on a single disk because the latter has only one I/O path.
- The `defragment` utility processes each volume in parallel. Thus, creating a domain on several volumes can increase the defragment speed.
- Combining multiple volumes within a single domain allows you to build larger domains and reduces the overall management effort because fewer domains require less administration. However, a single volume failure within a domain renders the entire domain inaccessible.
- Limiting the number of volumes decreases the risk of disk errors that can cause the entire domain to become inaccessible. To improve reliability, you can set up LSM mirrors. See Section 1.7 for information about setting up AdvFS with LSM. See Logical Storage Manager for LSM functionality.

The System Configuration and Tuning guide provides detailed guidelines for planning and configuring your file system. The AdvFS Software Product Description (SPD) contains information about system limits. The SysMan "Manage an AdvFS Domain" allows you to create and configure domains. See Chapter 6 if you want to configure your domains using the AdvFS GUI.

## 1.4.2 The `/etc/fdmns` Directory

The `/etc/fdmns` directory defines domains by providing a subdirectory for each domain you create. The subdirectories contain a symbolic link to every volume in the domain.

This directory is created and maintained automatically by AdvFS when you use standard AdvFS commands. You must have a current `/etc/fdmns` directory in order to mount the filesets in a domain.



When you create a domain, a soft link is created from the domain entry in the `/etc/fdmns` directory to the block device. You cannot create a domain simply by creating a link in the directory.

Back up the `/etc/fdmns` directory regularly. If the contents of the directory become corrupted or deleted, restore the directory from your most recent backup. A damaged directory prevents access to the domain because the information matching the domain to the physical volume containing the filesets is incorrect; the filesets are not affected.

If you attempt to mount a fileset from a domain with a damaged directory, a message similar to the following for the domain `accounts` will be displayed:

```
Volume count mismatch for domain accounts.  
accounts expects 2 volumes, /etc/fdmns/accounts has 1 links.
```

If you lose the `/etc/fdmns` directory or the volumes of a domain are moved to a different system, use the `advscan` utility to reconstruct the directory and to find the location of AdvFS domains on volumes or LSM disk groups (see Section 5.5.1).

### 1.4.3 Version 5 Domains

Version 5.0 and later operating system domains have an improved disk structure that provides support for quota values larger than 2 terabytes and increases performance for directories containing thousands of files. All domains that were created prior to Version 5.0 are recognized by later versions but are not automatically upgraded to the new structure.

A domain version number (DVN) is associated with a domain. The Version 5.0 operating system is the first for which a new domain carries a DVN of 4. Domains created earlier carry a DVN of 3.

You can mount a DVN3 fileset on a system running Version 5.0. You cannot locally mount a fileset in a DVN4 domain created under Version 5.0 or later (DVN4) on a system running a Version 4 operating system. If you need to access a DVN4 fileset, NFS mount it from a Version 5 or later server (see Section 5.1). Therefore, if your application requires backward compatibility, it is inadvisable to upgrade your domain to the new DVN.

See Section 5.1.1 for a discussion of utility incompatibility.

#### 1.4.3.1 Creating a Domain with DVN4

All domains created under Version 5.0 automatically have a DVN of 4. Therefore, if you do a full install of Version 5.0, all the domains created in the process will have the new structure.

If you update install Version 5.0, your existing domains will retain the DVN of 3. This means that `/root`, `/usr`, and `/var` will also have a DVN of 3. There is no conversion utility. To move your data to a DVN4 domain, back up your data, create a new DVN4 domain, and restore it to the new domain.

#### 1.4.3.2 Upgrading a Domain to DVN4

If you are running Version 5.0 and if you are running an application that requires quota limits larger than 2 terabytes or that uses directories containing thousands of files, you can improve performance by upgrading your old domain to a domain with a DVN of 4.

DVN4 domains automatically have an index created when the directory grows beyond a page, that is, about 200 files. Use the `showfile` command to determine if a directory is indexed. Choosing the `-i` option displays information about the index. See `showfile(8)` for more information.

To upgrade a domain, create a new domain on a Version 5.0 or later system and copy all the information from the old domain to it. If you are unfamiliar with creating domains, read Section 1.4.5 first. To upgrade a domain:

1. Back up the filesets in the domain to tape with the `vdump` command. It is a good idea to use the `-x` option for additional protection from `saveset` errors. You will lose all the data in your domain if you cannot restore it after creating a new domain.
2. Remove the old domain with the `rmfdmn` command. This will also remove the old fileset.
3. Create a new domain with the `mkfdmn` command. The new domain has a DVN of 4. Note that you do not need to use the `-x` and `-p` options for the `mkfdmn` command. Version 5.0 takes care of BMT allocations.
4. Create the new filesets with the `mkfset` command.
5. Restore the filesets in the new domain with the `vrestore` command.

For example, to upgrade the domain `domain_p` on `/dev/disk/dsk1c` containing filesets `fset_p` and `fset_m` and put them back on the same volume with the same names:

```
# vdump -0 -N -x 8 /fset_p
# vdump -0 -N -x 8 /fset_m
# umount /fset_p
# umount /fset_m
# rmfdmn domain_p
# mkfdmn /dev/disk/dsk1c domain_p
# mkfset domain_p fset_p
# mkfset domain_p fset_m
# mount domain_p#fset_p /fset_p
# mount domain_p#fset_m /fset_m
```

```
# vrestore -x -D /fset_p
# mt fsf 1
# vrestore -x -D /fset_m
```

It is possible to create domains with a DVN of 3, which would be recognized by older operating systems, on a Version 5 system. See `mkfdmn(8)` for more information.

## 1.4.4 Displaying Domain Information

If a domain is active (at least one fileset is mounted), you can display detailed information about the domain and the volumes included in it with the `showfdmn` command:

```
showfdmn domain_name
```

For example, to display domain information for the `domain_1` domain:

```
# showfdmn domain_1
```

	Id	Date Created	LogPgs	Version	Domain Name		
	2bb0c594.00008570	Fri Mar 24 12:33 2000	512	4	domain_1		
Vol	512-Blks	Free %	Used	Cmode	Rblks	Wblks	Vol Name
1L	8325	79210	90%	on	128	128	/dev/disk/dsk1c
2	832527	1684	98%	on	128	128	/dev/disk/dsk2c
-----							
	1665054	80894	94%				

## 1.4.5 Creating a Domain

The first step in setting up an AdvFS file system is creating a domain and assigning an initial volume to it. However, a domain is not a complete file system that you can mount. In order to mount an AdvFS file system, the domain must contain one or more filesets. You can access files as soon as you mount one or more filesets (see Section 1.5).

Creating a single-volume domain with a single fileset is equivalent to creating a traditional UFS file system. To set up an active, single-volume file system, as illustrated in Figure 1-2, use the SysMan “Create a New AdvFS Domain,” the AdvFS GUI or, from the command line, enter the `mkfdmn(8)` command:

```
mkfdmn volume_name domain_name
```

To create an active single-volume DVN4 domain:

1. Create a single-volume domain associated with a volume using the `mkfdmn` command.

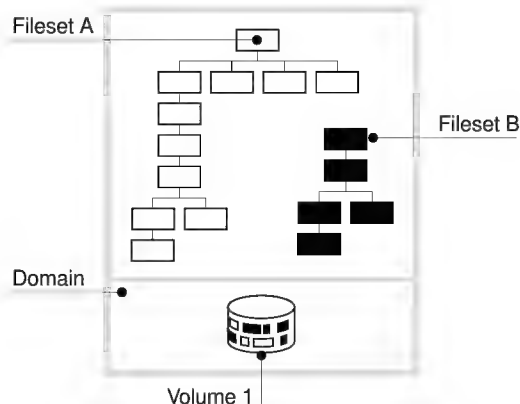
2. Create one or more filesets with the `mkfset` command and name each fileset the same as its mount-point directory; for example, if the mount-point directory is `/tmp`, name the fileset `tmp`. (This naming scheme is recommended but is not required.)
3. Create the mount-point directory with the `mkdir` command.
4. Mount each fileset with the `mount` command.

To create a domain of DVN3, see the `mkfdmn(8)` reference page.

### Caution

Do not use the `mkfdmn` command on a volume containing data that you want to keep. Doing so will destroy it. If you have accidentally used the `mkfdmn` command, you may be able to recover some of your data with the `salvage` utility if the volume that was destroyed was an AdvFS volume (see Section 5.4.6).

**Figure 1–2: Single-Volume Domain**



The following example creates a single-volume DVN4 domain, `domain_2`, and two filesets, `fset_a` and `fset_b`, in the domain. Because the domain has only one volume, the files in both filesets physically reside on one volume (this is allowed in AdvFS):

```
# mkfdmn /dev/disk/dsk2c domain_2
# mkfset domain_2 fset_a
# mkfset domain_2 fset_b
# mkdir /fset_a
# mkdir /fset_b
# mount domain_2#fset_a /fset_a
# mount domain_2#fset_b /fset_b
```

You can set up a domain with an LSM volume by naming the LSM volume in the `mkfdmn` command. See Section 1.7 for information about using AdvFS with LSM. If you have AdvFS Utilities, you can change the size of your domain by adding more volumes. You can transform a single-volume domain (except the root domain) into a multivolume domain (see Section 1.4.6 and Chapter 6).

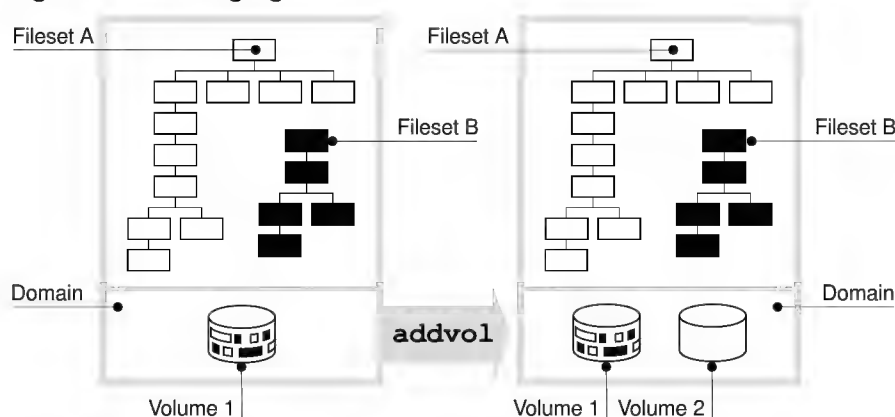
If you try to create a domain on a volume that is in use, you will get an error message. You can override the message and create the domain. For example, if `/dev/disk/dsk3g` is in use and you try to create the domain `usr_domain`, you will get the following message:

```
# mkfdmn /dev/disk/dsk3g usr_domain
Warning: /dev/rdisk/dsk3g is marked in use for 4.2BSD.
If you continue with the operation you can
possibly destroy existing data.
CONTINUE? [y/n] <y>
```

## 1.4.6 Increasing the Size of a Domain

You can expand a domain by replacing one of the volumes in the domain with a larger storage device or, alternatively, adding another volume to the domain. Figure 1-3 shows a graphic illustration of adding volumes to a domain. Neither adding nor removing volumes affects the directory hierarchy layer; all path names for the files remain the same. Also, the file system can remain active during the disk exchange.

**Figure 1-3: Enlarging a Domain**



If you plan to increase the size of a DVN3 domain, there may be performance benefits if you upgrade to the new file structure (see Section 1.4.3) before you do this.

---

**Caution**

---

If your domain is located on an LSM volume, do not use the LSM `grow` option to increase its size.

---

A newly created domain consists of one volume, which can be a disk, disk partition, or logical volume. To add volumes, use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `addvol` command:

```
addvol device_name domain_name
```

For example, to add volume `dsk3c` to the domain `resources`:

```
# addvol /dev/disk/dsk3c resources
```

You can add volumes immediately after creating a domain, or you can wait until the domain requires additional space. You can add a volume to an active domain while its filesets are mounted and in use.

You cannot add a volume to the root domain unless you are running a cluster (see Cluster Administration). If you are running a cluster configuration, adding another root volume is the same procedure as adding any volume.

---

**Caution**

---

Do not use the `addvol` command to add a volume containing data that you want to keep. Doing so will destroy it. If you have accidentally used the `addvol` command, you may be able to recover some of your data with the `salvage` utility if the volume that was destroyed was an AdvFS volume (see Section 5.4.6).

---

If you do not have AdvFS Utilities and want to increase the size of your domain, you must move the domain to a different volume with the `vdump` and `vrestore` commands process (see Section 4.3.3).

To increase the size of a domain if you have AdvFS Utilities:

1. Use the `showfdmn` command to display the contents of the domain and the current disk capacity of each volume. This step is optional.
2. Add the new volume to the domain.
3. Remove the old volume if you do not want it.
4. Run the `balance` utility to even the file distribution between the volumes. This step is not required.

The following example replaces one disk, the volume `/dev/disk/dsk2c`, of the `domain_1` domain with a larger disk, the volume `/dev/disk/dsk3c`:

```
# showfdmn domain_1
      Id           Date Created      LogPgs Version Domain Name
2bb0c594.00008570 Fri Mar 24 12:33 2000 512          4 domain_1

Vol  512-Blks  Free % Used Cmode Rblks Wblks Vol Name
  1L   832527 79210   90%   on   128   128 /dev/disk/dsk1c
   2   832527  1684   98%   on   128   128 /dev/disk/dsk2c
-----
      1665054 80894   94% # addvol /dev/disk/dsk3c domain_1
# rmvol /dev/disk/dsk2c domain_1
# balance domain_1
```

AdvFS will not add a volume that will cause partitions to overlap with a volume that is mounted for another file system, a swap area, or a reserved partition. To add an overlapping partition, unmount all filesets and use the `addvol` command with the `-F` option. Note that disabling the overlap check can result in extensive data loss and should be used with extreme caution (see `addvol(8)`).

## 1.4.7 Reducing the Size of a Domain

When there is sufficient free space on the remaining volumes, you can remove volumes from a domain without interrupting users or affecting the logical structure of the filesets in the domain. The file system automatically migrates the contents of the selected volume to other volumes in the domain. Before you can remove a volume from a domain, all filesets in the domain must be mounted. An error will occur if you try to remove a volume from a domain with unmounted filesets.

If there is not enough free space on other volumes in the domain to accept the files offloaded from the departing volume, as many files as possible are moved to available free space on other volumes. Then a message is sent indicating that there is insufficient space. The domain is not damaged.

---

### Caution

---

If your domain is located on an LSM volume, do not use the LSM `shrink` option to reduce its size.

---

To remove a volume, use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `rmvol` command:

```
rmvol device_name domain_name
```

For example, to remove `dsk3c` from the domain inventory:

```
# rmvol /dev/disk/dsk3c inventory
```

You can interrupt the `rmvol` process (see `rmvol(8)`) with `Ctrl/C` or with the `kill -term` command without damaging your domain. Do not use the `kill -KILL` command. Files already removed will remain in their new location.

Under some circumstances interrupting an `rmvol` command with the `kill` command can leave the volume in an inaccessible state. If a volume does not allow writes after an aborted `rmvol` operation, use the `chvol` command with the `-A` option to reactivate the volume.

To reduce the size of a domain:

1. Use the `showfdmn` command to display the contents of the domain and the current disk capacity of each volume. This step is optional.
2. Remove the volume.
3. Run the `balance` utility to even the file distribution between the remaining volumes. This step is not required.

You cannot remove a volume if you are balancing or defragmenting a domain.

The following example shows how to remove one disk of the `domain_2` domain, `/dev/disk/dsk2c`:

```
# showfdmn domain_2
  Id                Date Created      LogPgs Version Domain Name
2bb0c594.00008570 Fri Jun 9 10:23 2000 512      4 domain_2
  Vol 512-Blks   Free % Used Cmode Rblks Wblks  Vol Name
-----
  1L   832527   386984   54%   on   128   128 /dev/disk/dsk1c
  2    832527   647681   22%   on   128   128 /dev/disk/dsk2c
  3    832527   568894   32%   on   128   128 /dev/disk/dsk3c
-----
      249758 1603559   36%
# rmvol /dev/disk/dsk2c domain_2
# balance domain_2
```

If you remove an AdvFS volume that contains a stripe segment, the `rmvol` utility moves the segment to another volume that does not already contain a stripe segment of the same file. When a file is striped across all volumes in the domain, a confirmation is required before removing the volume. If you allow the removal process to continue, more than one stripe segment will be placed on the remaining volumes. See Section 4.3.4 for details on file striping.



### 1.4.8 Removing a Domain

You can remove a domain after all filesets in the domain are unmounted. When you remove a domain, the entry in the `/etc/fdmns` directory that defined the domain is removed and you cannot mount the filesets. Volumes that were assigned to the removed domains are relabeled as unused and can be reused.

Note that removing the domain to inactivate filesets instead of using the `rmfset` command to remove each fileset may present a security hole because no data on the volumes is changed. It may be possible to access the data with the `salvage` command (see Section 5.4.6).

To remove a domain, use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, unmount all filesets and clone filesets. Then, enter the `rmfdmn` command:

```
rmfdmn domain_name
```

For example, to remove the domain promotions:

```
# rmfdmn promotions
rmfdmn: remove domain promotions? [y/n]y
rmfdmn: domain promotions removed
```

If you attempt this command when there are mounted filesets, the system displays an error message. AdvFS will not remove an active domain.

### 1.4.9 Renaming a Domain

An existing domain can be assigned a new name without altering its domain identifier, the set of numbers that identifies the domain. When you rename a domain, entries for all filesets in the domain must be updated in the `/etc/fstab` file (see Section 1.5.2). To rename a domain, use the AdvFS GUI (see Chapter 6) or, from the command line:

1. Unmount all the filesets and any related clones.
2. In the `/etc/fdmns` directory, change the old domain name to the new one:  

```
mv /etc/fdmns/old_dom_name /etc/fdmns/new_dom_name
```
3. Edit the `/etc/fstab` file to enter the new domain name and remove the old.
4. Mount the filesets in the renamed domain.

For example, to rename the domain `marketing` to `advertising`, assuming one fileset, `fset`, is mounted at `/fset`:

```
# umount /fset
# mv /etc/fdmns/marketing /etc/fdmns/advertising
# vi /etc/fstab
```

Change the line

```
marketing#fset /fset advfs rw,userquota,groupquota 0 2
```

to

```
advertising#fset /fset advfs rw,userquota,groupquota 0 2
```

Mount the fileset.

```
# mount /fset
```

## 1.5 Filesets

A fileset represents a mountable portion of the directory hierarchy of a file system. Filesets and traditional UNIX file systems are equivalent in many ways: you mount AdvFS filesets, filesets contain files, are units on which you enable quotas, and are units for backing up data.

In contrast with traditional file systems, the directory hierarchy of AdvFS is independent of the storage. Therefore, you can change file placement without affecting the logical structure of the filesets.

Fileset names may be associated with their domain names as in `filedomain#fileset`. Here the pound sign (#) is part of the naming syntax and does not represent a comment.

Fileset names must be unique within a domain. Each fileset has a unique ID composed of the domain ID and a fileset tag. Each domain has its own series of fileset tags.

AdvFS also supports clone filesets. A clone fileset is a read-only copy of an existing fileset created to capture data at one instant in time (see Section 4.1.5).

### 1.5.1 Configuring Filesets

The filesets in a domain share the available space on the volumes in the domain and use the same domain transaction log. Each fileset has its own directory structure, root tag directory, quota files, and frag file. The optimal number of AdvFS filesets in a domain depends primarily on the requirements of the applications that use the filesets.

It is generally a good idea to have multiple filesets rather than one large one. Whether you place these filesets in a number of domains rather than in one large one depends on your application. If a fileset has critical access requirements, place it in its own domain. For applications that perform many

file create and remove operations, configure multiple filesets and distribute the files across the filesets. This will reduce I/O and related contention.

Create multiple smaller filesets rather than a few large ones if you do not want to share space between filesets. It is a good idea to place each or a few filesets in a single domain on its own volume.

Create filesets containing files having similar requirements in the same domain. For example, do not place small temporary mail news server files in the same domain as the database.

Because filesets are managed independently of their physical storage, each fileset can be backed up independently (see Chapter 3) and can be assigned its own quota limits (see Chapter 2). Multiple small filesets can be backed up and restored more quickly than a single large fileset. You can run the `vdump` or `vrestore` commands on several filesets simultaneously.

You can group files by their management requirements. For example, you could create a fileset for developer files that will be backed up twice a day and you could create another fileset with quotas imposed to limit the amount of disk space available to the marketing department.

Because the one transaction log in domain is shared by all filesets, filesets with a large amount of I/O can cause the transaction log to become a bottleneck (see Section 1.1.2). Balance the management gains of having multiple filesets in a domain against the potential performance reduction caused by having all of the log data for all filesets going to one transaction log (see Section 4.1.3).

You can create and mount filesets until the system runs out of system resources (such as memory or disk space). The Tru64 UNIX Software Product Description contains information about this and other system limits.

System Configuration and Tuning provides detailed guidelines and suggestions for file system configuration. The SysMan “Manage an AdvFS Domain” allows you to create and configure filesets. Many configuration operations can be accomplished with the AdvFS GUI (see Chapter 6).

## 1.5.2 Designating Filesets in the `/etc/fstab` File

AdvFS filesets are added to the `/etc/fstab` file by listing them with an `advfs` designation (see `fstab(4)`). This is similar to the way that you add any other file system. AdvFS filesets listed in the `/etc/fstab` file are mounted each time you reboot the system.

The fileset entry includes the domain name, fileset name, mount point, file system type, and the mount point options. The user quota and group quota options (see Section 2.2) should be included along with the `pass` field numbers if quotas are used. If they are not desired, the user quota and group

quota options can be omitted. An AdvFS `/etc/fstab` entry with user and group quotas enforced should include:

```
file_dmn#fileset /mount_point advfs rw,userquota,groupquota 0 2
```

For example, to automatically mount the `credit` fileset with user and group quotas (assuming the fileset's mount point exists), add the following line to the `/etc/fstab` file:

```
acct_domain#credit /usr/credit advfs rw,userquota,groupquota 0 2
```

### 1.5.3 Displaying Fileset Information

Any system user can display detailed information about mounted filesets and clones. Root user privilege is required only if the domain is inactive (filesets unmounted). To examine fileset information enter:

```
showfsets domain_name
```

The following example displays the domain `big_domain`, which has four filesets:

```
# showfsets big_domain
staff1_fs
  Id      : 2cb9d009.000419f4.1.8001
  Files   : 18554,  SLim= 0, HLim= 0
  Blocks(512) : 712230,  SLim= 0, HLim= 0
  Quota Status : user=on group=on
guest_fs
  Id      : 2cb9d009.000419f4.2.8001
  Files   : 4765,  SLim= 0, HLim= 0
  Blocks(512) : 388698,  SLim= 0, HLim= 0
  Quota Status : user=on group=on
staff2_fs
  Id      : 2cb9d009.000419f4.3.8001
  Files   : 12987,  SLim= 0, HLim= 0
  Blocks(512) : 842862,  SLim= 0, HLim= 0
  Quota Status : user=on group=on
staff3_fs
  Id      : 2cb9d009.000419f4.4.8001
  Files   : 48202,  SLim= 0, HLim= 0
  Blocks(512) : 1341436,  SLim= 0, HLim= 0
  Quota Status : user=on group=on
```

The following example displays `domain_2`, which contains one fileset and one clone fileset:

```
# showfsets domain_2
test_fs
  Id      : 3003f44f.0008ac95.4.8001
  Clone is : clone_test
  Files   : 7456,  SLim= 0, HLim= 0
```

```
Blocks (512) : 388698, SLim= 0, HLim= 0
Quota Status : user=on group=on
Clone_test
  Id          : 3003f44f.0008ac95.5.8001
  Clone of    : test_fs
  Revision    : 2
```

You can use the AdvFS GUI (see Chapter 6) to obtain similar information.

### 1.5.4 Creating a Fileset

A domain must contain at least one mounted fileset to be active (see Section 1.1.1). Any fileset can consume all of the storage available in the domain. You can also create multiple filesets within a domain that share the storage pool established for the domain.

Each fileset can be mounted and unmounted independently of the other filesets in the domain. You can limit fileset growth within a domain by assigning fileset quotas (see Chapter 2).

To create a fileset in a domain, use the SysMan “Create a New AdvFS Fileset,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `mkfset` command:

```
mkfset domain_name fileset_name
```

For example, to create the fileset `coupons` in the domain `advertising`:

```
# mkfset advertising coupons
```

See also Section 1.4.5, which contains examples for configuring an active (filesets mounted) domain.

### 1.5.5 Upgrading a Fileset

Filesets that are part of domains created for Version 5.0 and later (DVN4) support large quota values and have better performance for very large directories. Filesets in domains created earlier (DVN3) do not have these improvements. To upgrade a fileset to the new version, you must upgrade its domain (that is, create a new one) and restore the fileset to it. Then the restored fileset will have the new structure (see Section 1.4.3).

You cannot mount filesets with the new DVN on operating systems earlier than Version 5.0 unless you NFS mount them from a Version 5.0 or later server (see Section 5.1).

### 1.5.6 Mounting a Fileset

As with traditional UNIX file systems, AdvFS filesets must be mounted in order to access them. Filesets to be mounted must be compatible with the operating system on which they were created (see Section 5.1).

To mount a fileset, use the SysMan “Mount File Systems,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `mount` command, where *mnt\_point* is the path to the mount point:

```
mount domain_name#fileset_name mnt_point
```

For example, to mount the fileset `coupons` in the domain `advertising`:

```
# mount advertising#coupons /coupons
```

Before a fileset is mounted, AdvFS verifies that all data in all volumes in a domain can be accessed. If there are problems, the mount may fail or the fileset may be mounted as read-only (see Section 5.4.9).

If you attempt to mount a fileset with an incorrect number of volumes, the mount operation will fail. See Section 5.5.1.2 and `advscan(8)` for further information.

You will get an error message if you attempt to mount a fileset created under Version 5.0 on a system running Version 4 or earlier (see Section 5.1).

### 1.5.7 Unmounting a Fileset

If you unmount a fileset, the fileset remains in the domain but it is not accessible. Mount the fileset to make it available again.

To unmount an AdvFS fileset, use the SysMan “Dismount an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `umount` command:

```
umount mnt_point
```

For example, to unmount the fileset `coupons` in the domain `advertising` that was mounted in Section 1.5.6:

```
# umount /coupons
```

### 1.5.8 Removing a Fileset

If you remove a fileset, it cannot be remounted. It is no longer part of the domain.

You must unmount a fileset before you can remove it. The fileset’s clone must be removed first. If you have set up a trashcan directory (see Section 4.4) for the fileset, it is also removed.

To remove a fileset, use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `rmfset` command:

```
rmfset domain_name fileset_name
```

For example, to remove the `tmp_1` fileset in `domain_1`:

```
# rmfset domain_1 tmp_1
rmfset: remove fileset tmp_1? [Y/N]y
```

The fastest way to remove all filesets is to remove the domain with the `rmfdmn` command. However, the `salvage` utility (see Section 5.4.6) may be able to access some of the data. The `rmfdmn` command removes the definition of the domain in the `/etc/fdmns/` directory and relabels the volumes, but it does not touch any data on the volume. Filesets removed with the `rmfset` command are not recoverable with the `salvage` utility because the utility destroys pointers to the fileset data (metadata) and all the files in the fileset.

## 1.5.9 Renaming a Fileset

An unmounted fileset can be renamed. Filesets are known to the system by their fileset identifier, which is a combination of the domain identifier and an additional set of numbers that identify the fileset within the domain. The fileset name is kept within the domain and is an attribute that you assign. When you rename a fileset, only this assignment is changed. The fileset identifier is not altered.

You can use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI, which updates the `/etc/fstab` file (see Chapter 6) to rename a fileset or, from the command line, enter:

```
renamefset domain_name old_fileset_name new_fileset_name
```

After renaming the fileset, you must update the corresponding entries in the `/etc/fstab` file (see Section 1.5.2). If you do not do this, AdvFS will not mount the fileset when the system is booted.

For example, to rename the `public` fileset mounted at `/mntpt` to `private`:

```
# umount /mntpt
# renamefset dm1_1 public private
# vi /etc/fstab
```

Change the line

```
dm1_1#public /mntpt advfs rw,userquota,groupquota 0 2
```

to

```
dm1_1#private /mntpt advfs rw,userquota,groupquota 0 2
```

and mount the fileset

```
# mount /mntpt
```

---

**Note**

---

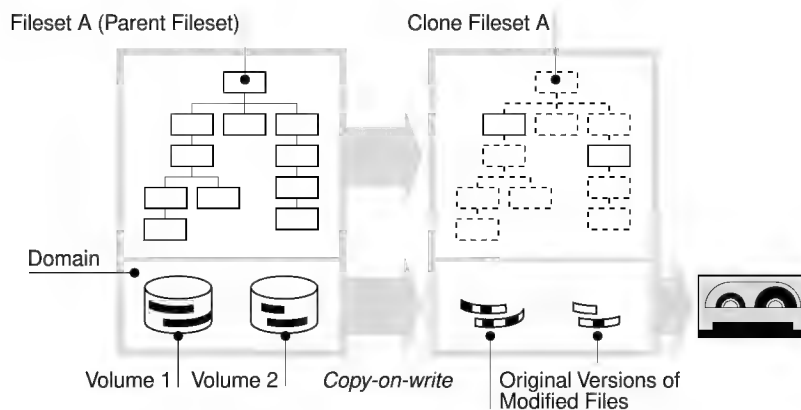
If you rename a fileset, its clone can no longer track it. You cannot rename the clone. You must delete the old clone and create a new one.

---

### 1.5.10 Clone Filesets

If you have the optional AdvFS Utilities, you can back up your files using a clone fileset, a read-only snapshot of fileset data structures (metadata). When you clone a fileset (create a clone fileset), the utility copies only the structure of the original fileset, not the actual data. Only when a file is modified does the file system copy the original, unchanged data to the clone fileset. (This is called copy-on-write.) Because the only data in the clone fileset is a copy of data that has been modified, the clone fileset is usually smaller than the original fileset. Figure 1-4 illustrates the relationship between parent and clone filesets.

**Figure 1-4: Cloning a Fileset**



Clone filesets increase the availability of data because they:

- Preserve the system at a particular time

A clone is not a replacement for the backup process (using the `vdump` command, for example). However, it can provide internal consistency if you are trying to back up a system where files are changing rapidly and you want to retain the information at a particular time. Use the clone to make a snapshot of your data, then use the `vdump` command to back up the clone.



- Protect against accidental file deletion or corruption  
Create a clone of each fileset that you plan to access or modify. By leaving the done fileset on line, you can replace unintentionally deleted or corrupted files without loading backup tapes.

See Section 3.2 for command line instructions on using cloning for online backup and Section 6.4.3 for an explanation of cloning with the AdvFS GUI.

---

**Note**

---

Changing text files with an editor may cause the entire original file to be copied to the clone. Many editors rewrite the entire file regardless of what has changed. When this happens, your clone fileset may grow very large. There is no way for AdvFS to alter this process.

---

When you delete a file that existed when the clone was created, it remains available (but not visible in the original fileset) for the life of the clone. The file is not copied to the clone, but the actual delete is delayed until the clone is deleted. The version of the file that is retained is the one that existed when the clone was created. Later updates are lost.

The size of the clone fileset depends upon the number of updates that occur during the life of the clone. The `df` command, which displays statistics on free disk space, does not accurately reflect the size of the clone fileset because it constantly changes as files are updated.

---

**Caution**

---

When a domain runs out of disk space, the file system loses its ability to maintain the consistency of files within clone filesets. The original fileset is usable, but the clone fileset is not accurate. A warning message is displayed on both the user's terminal and the system console.

---

#### 1.5.10.1 Creating a Clone Fileset

Cloning is transparent to the user and has little impact on system performance. You do not have to be root user to create a clone. To create a clone fileset, use the SysMan "Manage an AdvFS Domain," the AdvFS GUI (see Chapter 6) or, from the command line, enter the `clonefset` command:

```
clonefset domain_name fileset_name clone_name
```

For example, to create a clone called `clone_day300` for the fileset `day300` in the domain `transactions`:

```
# clonefset transactions day300 clone_day300
```

#### 1.5.10.2 Mounting and Unmounting a Clone

Clones are mounted and unmounted the same as any other fileset (see Section 1.5.6 and Section 1.5.7).

#### 1.5.10.3 Removing a Clone

Clones are removed the same as any other fileset (see Section 1.5.8).

#### 1.5.10.4 Renaming a Clone

A clone fileset cannot be renamed. To assign a new name, remove the old clone and create a new clone for the fileset. (Note that this new clone is a snapshot of the fileset at a later point in time than the deleted clone.)

If you have renamed a fileset that has a clone, you must delete the clone associated with the old name before you can create a clone for the newly named fileset.

## 1.6 Configuring an AdvFS root File System

There are several advantages to configuring the root file system on AdvFS. You can:

- Restart quickly after a crash. You do not run the `fsck` utility after a crash.
- Use one set of tools to manage all local file systems. All features of AdvFS except `addvol` and `rmvol` are available to manage the root file system.
- Use AdvFS with LSM to mirror the root file system. This allows your root file system to remain viable even if there is a media failure.

The following restrictions on the AdvFS root file systems are currently enforced:

- Unless you are running a cluster (see Cluster Administration), the root domain can only contain one volume. You cannot add volumes to the root domain.
- The volume must start from the beginning of the physical device (a or c partitions).
- The root fileset must be the first fileset created in the root domain.
- You can assign any name to the root domain and fileset but the same name must be entered in the `/etc/fstab` file.

You can put the root file system on AdvFS during the initial base-system installation or you can convert your existing root file system after installation. Note that when you install AdvFS as root during the initial installation, root will default to the `a` partition.

If you construct your own root file system, you can configure it on the `a` or `c` partition. See Section C.2 for instructions on converting an existing UFS root file system to AdvFS. See the Installation Guide for instructions on installing AdvFS as the root file system during the initial installation.

### 1.6.1 Mounting the root File System in Single-User Mode

The root file system is automatically mounted as read-only when the system is booted in single-user mode. You can change the root fileset mount from read-only to read-write with the `mount -u` command:

```
mount -u /
```

Use this procedure when you need to make modifications to the root configuration. For example, use it if you need to modify your `/etc/fstab` file.

### 1.6.2 Changing the Name of the root Domain

You can change the name of the root domain the same as any other domain (see Section 1.4.9). The name of a root domain is stored as the directory name in the `/etc/fdmns` directory and in the entry for root in the `/etc/fstab` file.

### 1.6.3 Changing the Name of the root Fileset

Changing the name of the root fileset is similar to changing the name of any other fileset (see Section 1.5.9). There are, however, two complications:

- The `renamefset` command requires that the fileset be unmounted, and you cannot unmount the root fileset.
- You must edit the `/etc/fstab` file to change the name of the root fileset. To do this, you must make the root fileset writable. But the root fileset cannot be mounted as writable unless the `/etc/fstab` entry is correct.

Therefore, you must use an alternate bootable partition and manipulate the root fileset you are changing as you would an ordinary fileset, make the changes, then reboot the changed fileset as root.

To rename the root fileset:

1. Boot a partition other than the one you want to change. (It can be UFS.)

2. Make a new entry in the `/etc/fdmns` directory of the booted partition for the fileset whose name you want to change.
3. Change to the new directory and make a symbolic link to the device holding the original fileset.
4. Use the `renamefset` command to rename the root fileset.
5. Mount the newly named root fileset at a temporary location in order to update its `/etc/fstab` file.
6. Change the `fstab` entry to correspond to the new root fileset name.
7. Shut down the alternate system.
8. Reboot the original AdvFS system.

The following example changes the name of the root fileset from `root_fs` to `new_root`. Assume that the root fileset is in the `root_domain` domain on `/dev/disk/dsk2a`.

1. Boot a device other than the one you want to change.
2. Make an entry for `tmp_root_domain` in the `/etc/fdmns` directory:

```
# mkdir /etc/fdmns/tmp_root_domain
```

3. Change to the new directory and make a symbolic link for `tmp_root_domain`:

```
# cd /etc/fdmns/tmp_root_domain
# ln -s /dev/disk/dsk2a
```

4. Rename the fileset from `root_fs` to `new_root`:

```
# renamefset tmp_root_domain root_fs new_root
```

5. Mount the changed root to update the associated `fstab` file:

```
# mount tmp_root_domain#new_root /mnt
```

6. Edit the `/mnt/etc/fstab` entry for `tmp_root_domain`:

```
# cd /mnt/etc
# vi fstab
```

Change the line:

```
root_domain#root_fs / advfs rw,userquota,groupquota 0 2
to:
```

```
root_domain#new_root / advfs rw,userquota,groupquota 0 2
```

7. Shut down the alternate system.

```
# shutdown -h now
```

8. Reboot the AdvFS system.

---

**Note**

---

If you change the root domain and fileset names and forget to change the `/etc/fstab` entries, you will not be able to boot past single-user mode. You will have to fix `/etc/fstab` from single-user mode using an editor before you can proceed.

---

## 1.7 Using AdvFS with LSM

LSM is a disk-management tool that allows you to create arbitrary-sized volumes independent of disk sizes and partition boundaries. You can create mirrored and striped volumes and change their attributes as performance and availability needs dictate. For more information see Logical Storage Manager.

AdvFS treats LSM volumes just like any other volume such as a disk or a disk partition. You can use LSM to create and manage the volumes that you use in AdvFS domains.

LSM and AdvFS together provide some performance enhancements:

- Availability  
Volume mirroring and RAID allow access to the same information on more than one volume. This provides backup in the case of a disk failure.
- I/O performance  
Mirroring can improve the read throughput because files can be accessed from either volume depending upon I/O load.  
LSM's volume striping is useful when large files will be shared and when the transaction log is spread over multiple disks. AdvFS can only stripe individual files.
- Performance monitoring  
Detailed information on disk I/O activity is available with LSM disk monitoring.

Use AdvFS with or without LSM volumes to manage file systems and file-level activities:

- Create domains and filesets.
- Expand and shrink domains.
- Perform online backups.
- Set quotas on users, groups, and filesets.
- Configure and maintain file systems online.

- Stripe individual files rather than all files on a volume. Do not stripe individual files when using LSM striped volumes.
- Recover from system failures quickly.

You can operate on LSM volumes running AdvFS either through the command line or the AdvFS GUI (see Chapter 6). There is also an LSM GUI that manipulates LSM volumes. See Logical Storage Manager for further information.

### 1.7.1 Setting Up AdvFS with LSM Volumes

To use LSM with AdvFS from the command line:

1. Create the LSM volumes with the desired attributes.  
You can specify mirrored volumes, striped volumes, mirrored and striped volumes, volume location, and volume size.
2. Create a domain with the `mkfdmn` command and identify the LSM volume as the initial volume.
3. To create a multivolume domain, use the `addvol` command with either an LSM or non-LSM volume.

---

#### Caution

---

Do not use the `grow` or `shrink` LSM options to change the size of AdvFS domains. Use the `addvol` and `rmvol` commands.

---

You will get an error message if you try to create an AdvFS domain on an LSM volume that is already opened or on a volume that is already labeled as in use by either UFS or AdvFS.

If you already have an AdvFS domain, you can encapsulate the domain into LSM using the encapsulation tools.

If mirrored or striped LSM volumes are part of an AdvFS domain that also includes non-LSM volumes, you do not have control over which files go to the mirrored or striped LSM volumes. To place specific data on mirrored or striped volumes, create an AdvFS domain that contains only LSM volumes with the attributes that you want. Then, put the files you want mirrored or striped in that domain.

The `showfdmn` command and the AdvFS GUI (see Chapter 6) include LSM volumes in the domain information display. In addition, you can use the `advscan` command to locate AdvFS volumes in LSM disk groups.

It is better to add multiple small LSM volumes rather than a single, large, striped or concatenated volume to an AdvFS domain. This enables AdvFS to balance and stripe I/O across volumes if necessary.

Note that you must use the full LSM path name when referencing an LSM volume.

The following example creates a domain on an existing LSM volume:

```
# mkfdmn /dev/vol/rootdg/vol01 domain_rs
```

You can also add volumes to an existing domain:

```
# addvol /dev/vol/rootdg/vol02 domain_rs
```

The following example creates a 1-gigabyte domain with two LSM volumes:

```
# volassist make vol01 500m
# volassist make vol02 500m
# mkfdmn /dev/vol/rootdg/vol01 onegb_domain
# addvol /dev/vol/rootdg/vol02 onegb_domain
# mkfset onegb_domain onegb_fset1
# mkdir /fset1
# mount onegb_domain#onegb_fs et1 /fset1
```

## 1.7.2 Mirroring LSM Volumes Under AdvFS

You can place two (or more) LSM volumes in a mirror configuration for AdvFS domains. This configuration provides distributed processing and fail-over protection; since both disks contain the same information, either one can be accessed. Logical Storage Manager provides detailed instructions for creating LSM mirrored volumes.

The following example creates a 500-megabyte mirrored LSM volume with a domain and one mounted fileset:

```
# volassist make mirr_vol 500m nmirror=2
# mkfdmn /dev/vol/rootdg/mirr_vol domain1
# mkfset domain1 fset_1
# mount domain1#fset_1 /mnt
```

To back up the LSM volume, split the mirror. Dissociate a plex from the volume, create a temporary volume with the plex, then start the volume:

```
# volplex dis mirr_vol-02 volplex
# volmake -Ufsgen vol temp_mirr_vol plex=mirr_vol-02
# volume start temp_mirr_vol
```

To mount the temporary volume, use the `mount` command with the `dual` option. If you do not use the `dual` option, you will get the error message:

```
E_DOMAIN_ALREADY_EXISTS
```

To mount the temporary volume:

```
# mkdir /bkup_fset
# mkdir /etc/fdmns/domain2
# ln -s /dev/vol/rootdg/temp_mirr_vol \
    /etc/fdmns/domain2/ temp_mirr_vol
# mount -o dual domain2#fset_1 /bkup_fset
```

Back up the temporary volume:

```
# vdump -0 -f /dev/tape/tape0 /bkup_fset
```

Remove the temporary volume and reattach the plex:

```
# umount /backup
# rm /etc/fdmns/domain2/temp_mirr_vol
# rmdir /etc/fdmns/domain2
# volplex -f dis mirr_vol-02
# voledit -rf rm temp_mirr_vol
# volplex att mirr_vol mirr_vol-02 &
```

### 1.7.3 Striping LSM Volumes and Striping AdvFS Files

You can distribute I/O across multiple volumes by striping. AdvFS and LSM striping both work well on a busy system. LSM is better suited for general striping, while AdvFS striping allows the administrator finer-grained control for positioning individual files.

When you choose LSM striping, you preconfigure your volumes for file striping; then all files located on the volumes configured for striping will be striped. You can do this from the command line or from the LSM GUI (see Logical Storage Manager).

You can stripe an AdvFS file at any time (see Section 4.3.4). Therefore, if you find that a file has become large or has heavy I/O requirements, you can use the AdvFS `stripe` command to stripe it.

It is not advisable to do both LSM and AdvFS striping because this may degrade system performance.

The following example creates a striped LSM volume and then creates an AdvFS domain using that striped LSM volume. All files created in this AdvFS domain will be striped across LSM volumes:

```
# volassist make vol06 600m layout=stripe nstripe=3
# mkfdmn /dev/vol/rootdg/vol06 striped_domain
```



---

## Managing Quotas

AdvFS allows you to activate quotas to track and control the amount of physical storage that each user, group, or fileset consumes. You must have root user privilege to set and edit quotas. The root user is never restricted by quota limits; quota limits only apply to nonroot users.

The AdvFS quota system is compatible with the Berkeley-style quotas of UFS. However, the AdvFS quota system differs in two ways:

- AdvFS differentiates between quota maintenance and quota enforcement. Quota information is always maintained, but enforcement can be activated and deactivated.
- AdvFS supports fileset quotas; that is, you can set quota limits for the filesets in a domain.

### 2.1 Introducing Quotas

You can set two quota values: on the amount of disk storage and on the number of files. In addition there are two types of quotas you can set:

- User and group quotas

AdvFS user and group quotas are similar to UFS quotas. You can set a separate quota for each user or each group of users for each fileset.

- Fileset quotas

You can restrict the space that a fileset itself can use. Fileset quotas are useful when a domain contains multiple filesets. Without fileset quotas, any fileset can grab all of the disk space in the domain.

For example, it is useful to set quotas on filesets that contain home directories such as `/usr/users` because these filesets can grow rapidly. Conversely, setting quota limits on the `/tmp` fileset is not recommended because this fileset is likely to fluctuate in size.

#### 2.1.1 Quota Limits

Limits are set on disk usage (number of blocks) or on number of files (inodes) or both. Table 2-1 shows the size limits for both types of quota values.

**Table 2–1: Quota Size Limits**

	User and Group Quotas	Fileset Quotas
Disk Usage	8 billion terabytes*	4 billion terabytes
Number of Files	4 billion	4 billion

\* Prior to Version 5.0, the disk usage limit for user and group quotas was 2 terabytes. If your domain was created prior to Version 5.0, and if you are now running Version 5.0 or later, you must upgrade your domain if you wish to take advantage of the new limits (see Section 1.4.3).

All quotas can have two types of limits: hard and soft. A hard limit cannot be exceeded. No more space can be allocated or files created. A soft limit permits a period of time during which the limit can be exceeded as long as the hard limit has not been exceeded.

Hard and soft limits can be set or changed by the root user at any time and take effect the next time quotas are activated. Hard and soft limits can be set for users, for groups, and for filesets.

The default is no quota limit. You can also:

- Set the hard limit to 0 blocks or files to impose no quota limits.
- Set the hard limit to 1 block or file to permit no disk space allocations.
- Set the soft limit to 1 block or file and the hard limit to 0 blocks or files to permit disk-space allocations on a temporary basis. These limits remain in effect until you unmount the fileset.

See Section 5.4.4 for directions if your activities cause you to run into limits.

## 2.1.2 Grace Period

Associated with each soft limit is a grace period. The grace period is the amount of time during which the soft limit can be exceeded. When the grace period expires, no one can create new files or allocate any more disk space until enough files are deleted to fall below the soft limit. Updating existing files may cause loss of data.

The timer for the grace period starts when the user exceeds the soft limit. If you allow no grace period, the user can exceed the soft limit only once. The grace period is turned off and reset each time usage drops below the soft limit. If you change the grace period after the user has exceeded the soft limit, the old grace period stays in effect until usage drops below the limit.

You can set grace periods for:

- Users

For each fileset you can set only one grace period for all users (see Section 2.2.2). If the user grace period expires, the user cannot allocate storage in the fileset until enough files are deleted to fall below the soft limit. The grace periods for the number of blocks and for the number of files do not need to be the same.

- Groups

For each fileset you can only set one grace period for all groups (see Section 2.2.2). If the group grace period expires, no one in the group can allocate storage in the fileset until enough files are deleted to fall below the soft limit. All users are limited by the group grace period, even if the fileset grace period is larger.

- Filesets

The fileset grace period is the same as the group grace period (see Section 2.3.2.2). Any user or group can cause the fileset to exceed its soft limit and thereby evoke the fileset grace period. Any user or group (not necessarily the one that exceeded the limit) can delete files to bring the fileset below the soft limit.

AdvFS sets a default grace period of 7 days. This period can be changed (see Section 2.2.2 and Section 2.3.2). You can specify the grace period in days, hours, minutes, or seconds. You can also:

- Set the grace period to 0 days to impose the default grace period of 7 days.
- Set the grace period to 1 second to allow no grace period.

## 2.2 User and Group Quotas

The following sections describe quota files and how quotas and grace periods are set for users and groups.

### 2.2.1 Quota Files

AdvFS creates quota files to track quotas, grace periods, and fileset usage. Quota files are maintained within the fileset but, unlike UFS, the user cannot delete or create them. Quota files are present in the fileset even if quota limits have not been established.

AdvFS keeps user and group quota information in the root directory of the fileset in the `quota.user` and `quota.group` files. These files are created when the fileset is created. They are indexed by user ID and group ID. Each quota file entry contains the following information: hard block limit, soft

block limit, block usage, hard file limit, soft file limit, file usage, block grace period, and file grace period.

Quota files are sparse files (see Section 1.1.3); that is, there are holes in the file where no user IDs or group IDs fall. Use the `ls` command with the `-l` option to see the space spanned by the `quota.user` or the `quota.group` file. This does not show the actual disk usage. For example:

```
# ls -l quota.user
-rw-r----- 1 root      operator  294912 Jul 20 08:50 quota.user
```

Use the `ls` command with the `-s` option to display how many blocks the file actually uses:

```
# ls -s quota.user
16          quota.user
```

To enforce user and group quotas for a fileset, the `/etc/fstab` file must have the following quota mount point options for the fileset:

```
advfs rw,userquota,groupquota 0 2
```

You can relocate the `quota.user` and `quota.group` files to subdirectories of the fileset. However, you cannot relocate them to other filesets nor delete them. If you relocate your files, you must update the `/etc/fstab` file entry to include the path and name of the relocated file(s) in the `userquota` and `groupquota` parameters.

For example, to relocate the `quota.user` file to the subdirectory `d4` and rename it `nquot`, change the `/etc/fstab` entry to:

```
dom_1#fset /mnt advfs rw,userquota=/mnt/d4/nquot,groupquota 0 2
```

In this example, the group quota file is not moved.

Whether or not you relocate your quota files, you do not need to enforce quotas. See Section 2.2.7 for instruction on deactivating user and group quotas.

## 2.2.2 Setting Quotas and Grace Periods

You are not required to set both user and group quotas. If you specify a group quota, it will apply to all users belonging to that group. If you specify a user quota that is larger than the group quota, it will have no effect because the group quota will take effect before the user quota is reached.

Use the `edquota` command to set the quotas for users and groups and again to change the grace period. The fileset must be mounted before you can set quotas and grace periods. Note that for each fileset, only one grace period can be set for all users and one grace period can be set for all groups. However, you do not have to set the same grace period for the number of

blocks and for the number of files. The group grace period you set is also entered as the fileset grace period (see Section 2.3.2.2).

Follow these general steps to set user and group quotas and grace periods:

1. Add quota file mount point options to the `/etc/fstab` file.
2. Enter the `edquota` command with the `-u` option to set user quotas or the `-g` option to set group quotas. An ASCII representation of the quota file is displayed through the editor specified by the `EDITOR` environmental variable. If `EDITOR` is not set, the `vi` editor is used.
3. When the user or group quota information is displayed, modify the values in the limits fields as needed. Then, exit the editor, saving the changes.
4. To set user or group grace periods, enter the `edquota` command with the `-ut` option for user or `-gt` option for group. Note that the fileset grace period is the same as the group grace period.
5. When the grace period information is displayed, modify the grace period as desired. Then, exit the editor, saving the changes.

To change the default editor and for more information, see `edquota(8)`.

The quota limits you set for a fileset do not take effect until you activate them. Use the `quotaon` command to activate quotas (see Section 2.2.5 and Section 2.2.6). If you are changing quota limits and quotas have already been activated for a fileset, the new limits become effective immediately.

If you have set quotas for a single user, you can use the `edquota` command with the `-p` option to create prototype quota files. Then you can apply the prototype quota to other users that you specify (see Section 2.2.3).

### 2.2.2.1 User Quotas Example

The following example sets quotas for the user `user5`:

1. If they do not already exist, add quota mount-point options to the `/etc/fstab` file. Note that there can be no spaces in the list of options delimited by commas; that is, from `rw` through `groupquota`:  

```
domain_1#test1 /test1 advfs rw,userquota,groupquota 0 2
domain_2#test3 /test3 advfs rw,userquota,groupquota 0 2
domain_4#test4 /test4 advfs rw,userquota,groupquota 0 2
```
2. Enter the `edquota` command with the `-u` option followed by the user name. If you specify more than one user name, the edits will affect all users named. The command creates a temporary file with an ASCII representation of the current quotas assigned to the named users and invokes an editor to allow you to modify the file:

```
# edquota -u user5
Quotas for user user5:
/test1: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
/test3: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
/test4: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

The values for `blocks in use` and `inodes in use` are the current block usage and the number of files for each fileset. You cannot change them. Soft and hard limits of 0 (zero) indicate that no limits have been set.

3. To change user quotas for `user5` for fileset `test3`, edit the file. Enter the new limits for disk usage on the `blocks` line and enter the new limits for the number of files on the `inodes` line:

```
/test3:blocks in use: 0, limits(soft=5000, hard=10000)
        inodes in use: 0, limits(soft= 100, hard= 200)
```

4. Exit the editor, saving the changes.

If quotas have already been activated for fileset `test3`, the new limits become effective immediately. If quotas are not yet activated for the fileset, the limits become effective as soon as quotas are activated (see Section 2.2.6).

Quotas set for the group to which the user belongs take precedence over quotas set for the user of the fileset.

### 2.2.2.2 User Grace Period Example

When you impose soft limits for a fileset, you can set a grace period for that fileset. If you do not specify a grace period, the grace period remains at the AdvFS default of 7 days. You can set different grace periods for the number of blocks and for the number of files. If the group grace period is less than a user grace period, the user is limited by the group grace period.

The following example sets the grace period for all users of fileset `test3`:

1. Run the `edquota` command with the user grace period options. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file:

```
# edquota -ut
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/test1: block grace period: 7 days,file grace period: 7 days
/test3: block grace period: 7 days,file grace period: 7 days
/test4: block grace period: 7 days,file grace period: 7 days
```

2. To set the user grace period for the number of blocks and for the number of files for `test3`, edit the file to change the existing grace period:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/test1: block grace period: 7 days,file grace period:7 days
/test3: block grace period: 2 days,file grace period:3 days
/test4: block grace period: 7 days,file grace period:7 days
```

3. Exit the editor, saving the changes.

If quotas have already been activated for the fileset `test3`, the new grace period becomes effective immediately. If a user has already exceeded the soft limit, the grace period becomes effective once the usage drops below the soft limit. If quotas are not yet activated for the fileset, the grace period becomes effective as soon as quotas are activated (see Section 2.2.6).

### 2.2.2.3 Group Quotas Example

The following example sets quotas on `test3` for the group `rsgusers`:

1. If they do not already exist, add quota mount-point options to the `/etc/fstab` file. Note that there can be no spaces in the list of options delimited by commas; that is, from `rw` through `groupquota`:

```
domain_1#test1 /test1 advfs rw,userquota,groupquota 0 2
domain_2#test3 /test3 advfs rw,userquota,groupquota 0 2
domain_4#test4 /test4 advfs rw,userquota,groupquota 0 2
```

2. Enter the `edquota` command with the `-g` option. If you specify more than one group name, the edits will affect all groups named. The command creates a temporary file with an ASCII representation of the current quotas assigned to the named groups and invokes an editor to allow you to modify the file:

```
# edquota -g rsgusers
Quotas for group rsgusers:
/test1: blocks in use: 0, limits (soft=0, hard=0)
        inodes in use: 0, limits (soft=0, hard=0)
/test3: blocks in use: 0, limits (soft=0, hard=0)
        inodes in use: 0, limits (soft=0, hard=0)
/test4: blocks in use: 0, limits (soft=0, hard=0)
        inodes in use: 0, limits (soft=0, hard=0)
```

The values for `blocks in use` and `inodes in use` are the current block usage and the number of files for each fileset. You cannot change them. Soft and hard limits of 0 (zero) indicate that no limits have been set.

3. To change the group quotas for `test3`, edit the file. Enter the new limits for disk usage on the `blocks` line and enter the new limits for the number of files on the `inodes` line:

```

/test3: blocks in use:0,limits(soft=60000, hard=80000)
        inodes in use:0,limits(soft= 6000, hard= 8000)

```

4. Exit the editor, saving the changes.

If quotas have already been activated for fileset `test3`, these limits become effective immediately. If quotas are not yet activated for `test3`, these limits become effective as soon as quotas are activated (see Section 2.2.6). Quotas set for the group take precedence over quotas set for the individual.

#### 2.2.2.4 Group Grace Period Example

When you impose soft limits, you can set one grace period per fileset for all groups. If you do not specify a grace period, the grace period remains the AdvFS default of 7 days. You can set different grace periods for the number of blocks and for the number of files. The group grace period takes precedence over all user grace periods.

The following example sets the grace period for all groups for fileset `test3`:

1. Run the `edquota` command with the group grace period options. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file:

```

# edquota -gt
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/test1: block grace period: 7 days,file grace period:7 days
/test3: block grace period: 7 days,file grace period:7 days
/test4: block grace period: 7 days,file grace period:7 days

```

2. To set the group grace period for the number of blocks and for the number of files for `test3`, edit the file to change the existing grace period:

```

Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/test1: block grace period: 7 days,file grace period:7 days
/test3: block grace period:12hours,file grace period:5 days
/test4: block grace period: 7 days,file grace period:7 days

```

3. Exit the editor, saving the changes.

If quotas have already been activated, this grace period becomes effective immediately unless a group has already exceeded the soft limit for `test3`. In that case, the new grace period becomes effective for that group once the group usage drops below the soft limit. If quotas are not yet activated, the group grace period becomes effective as soon as they are activated (see Section 2.2.6).



## 2.2.3 Setting Quotas for Multiple Users and Groups

AdvFS allows you to use a single command to modify quotas for a list of users or groups so you do not need to access and enter values for each one individually. Note that you do not have to set multiple grace periods because, for each user or group quota, one grace period per fileset applies to all users or to all groups.

Use the `edquota` command with the `-p` option to take existing quota information and establish it as a prototype user or group quota. Then apply the prototype to one or more users or groups.

For example, to set up all student accounts to have the same disk usage quota:

1. Establish a set of quotas for a single student. Use the `edquota` command with the desired limits.
2. Use the `edquota` command with the `-p` option to apply the quotas set up for the first user to other student accounts.

### 2.2.3.1 Prototype User Example

The following example sets up prototype-user quotas that are then used to modify the quotas for other users:

1. Set quotas for one user, `user5` (see Section 2.2.2.1):

```
# edquota -u user5
Quotas for user user5:
/test1:blocks in use:0,limits(soft= 20000,hard= 30000)
        inodes in use:0,limits(soft=   350,hard=   500)
/test3:blocks in use:1,limits(soft= 30000,hard= 40000)
        inodes in use:4,limits(soft=   400,hard=   550)
/test4:blocks in use:2,limits(soft= 10000,hard= 20000)
        inodes in use:1,limits(soft=   150,hard=   200)
/test5:blocks in use:2,limits(soft=100000,hard=150000)
        inodes in use:1,limits(soft=   5000,hard=   7000)
```

2. To create quotas for new users `user7`, `user8`, and `user9`, use the quotas from user `user5` as a prototype:

```
# edquota -p user5 -u user7 user8 user9
```

3. To verify that the quotas were set, run the `edquota` command for `user7`:

```
# edquota -u user7
Quotas for user user7:
/test1:blocks in use:0,limits(soft= 20000,hard= 30000)
        inodes in use:0,limits(soft=   350,hard=   500)
/test3:blocks in use:0,limits(soft= 30000,hard= 40000)
        inodes in use:0,limits(soft=   400,hard=   550)
```

```

/test4:blocks in use:0,limits(soft= 10000,hard= 20000)
          inodes in use:0,limits(soft=   150,hard=   200)
/test5:blocks in use:0,limits(soft=100000,hard=150000)
          inodes in use:0,limits(soft=  5000,hard=  7000)

```

### 2.2.3.2 Prototype Group Example

The following example sets up prototype group quotas that are then used to modify the quotas for another group:

1. Set quotas for the group `rsgusers` (see Section 2.2.2.3):

```

# edquota -g rsgusers
Quotas for group rsgusers:
/test1:blocks in use:0,limits(soft=100000,hard=200000)
          inodes in use:0,limits(soft= 10000,hard= 20000)
/test3:blocks in use:0,limits(soft=300000,hard=400000)
          inodes in use:0,limits(soft= 30000,hard= 40000)
/test4:blocks in use:0,limits(soft=500000,hard=600000)
          inodes in use:0,limits(soft= 50000,hard= 60000)
/test5:blocks in use:0,limits(soft=350000,hard=450000)
          inodes in use:0,limits(soft= 35000,hard= 45000)

```

2. To create quotas for a new group, `rsgstudents`, use the quotas from group `rsgusers` as a prototype:

```
# edquota -p rsgusers -g rsgstudents
```

3. To verify that the quotas were set, run the `edquota` command for `rsgstudents`:

```

# edquota -g rsgstudents
Quotas for group rsgstudents:
/test1:blocks in use:0,limits(soft=100000,hard=200000)
          inodes in use:0,limits(soft= 10000,hard= 20000)
/test3:blocks in use:0,limits(soft=300000,hard=400000)
          inodes in use:0,limits(soft= 30000,hard= 40000)
/test4:blocks in use:0,limits(soft=500000,hard=600000)
          inodes in use:0,limits(soft= 50000,hard= 60000)
/test5:blocks in use:0,limits(soft=350000,hard=450000)
          inodes in use:0,limits(soft= 35000,hard= 45000)

```

## 2.2.4 Verifying File and Disk Space Usage

Even if you are not enforcing quotas, you can monitor file and disk space usage with the `df`, `showfdmn`, `showfsets` with the `-k` option (to display in kilobytes), and `vdf` commands. See the appropriate reference pages.

If you are enforcing quotas, you can periodically verify your quota setup. You can display user and group quota information in a number of ways. If you are not the root user, you can display information only for your own files.

The root user can display all user and all group quota information for all filesets. All disk quota values are displayed in 1-kilobyte blocks.

The commands shown in Table 2-2 are useful for examining disk space and file usage for filesets for which user and group quotas are enforced.

**Table 2-2: User and Group Usage Commands**

Command	Description
<code>ncheck</code>	Prints the tag and full path name for each file in the fileset
<code>quot</code>	Summarizes fileset ownership
<code>quota</code>	Displays disk usage and limits by user or group
<code>quotacheck</code>	Checks fileset quota consistency
<code>repquota</code>	Summarizes quotas for a fileset

See the command reference pages for more information.

If your disk usage information appears to be corrupt, see Section 5.4.1 for instructions on how to correct this.

#### 2.2.4.1 Printing the Tag and Full Path Name for Each File

The `ncheck` command lists files by tag (inode) number. By piping the output to the `sort` command, you can use the sorted output as input for the `quot` command to list all files and their owners. Use the following format to generate the listing:

```
ncheck domain#fileset | sort +0n | quot -n domain#fileset
```

#### 2.2.4.2 Summarizing Fileset Ownership

The `quot` command displays block usage and the number of files in the fileset that each user owns. If you do not specify a fileset, the command processes all filesets in the `/etc/fstab` file that include the `ro`, `rw`, and `rq` mount options.

The `quot` command entered with no options displays only blocks:

```
# quot domain_1#set_1
domain_1#set_1:
34128    root
 816    user5
```

The `quot` command entered with the `-f` option displays both blocks and files:

```
# quot -f domain_1#set_1
domain_1#set_1:
34128      125    root
   816        9   user5
```

### 2.2.4.3 Displaying Disk Usage and Limits

The `quota` command displays the block usage, number of files, and quotas for a user or group. Users can run this command to look at their own disk space usage. The root user can look at usage for the whole system.

You can choose to display quota information for users or groups, for all filesets with usage over quota, or for all mounted filesets regardless of whether quotas are activated.

The `quota` command displays the block usage of the fileset, soft limit (`quota`), hard limit (`limit`), grace period, and number of files used for each user. An asterisk (\*) in a column means that a soft quota limit has been exceeded. Note that the grace period is not displayed unless the soft limit has been exceeded. Use the `edquota` command to view complete quota specifications.

The following example shows quota information for the user `user5`:

```
# quota -u user5
Disk quotas for user user5 (uid 446):
Filesystem  blocks  quota  limit grace files quota limit  grace
/           60    100    150           3    10    20
/usr       5071*  5000   10000 24:40      2    20    40
/test1      816   20000  30000           9   350   500
/test2     22032  50000 200000           2   2000  4000
/test3      2344  10000   15000        370  1000  2000
/test4     18023* 10000   20000 7days       3    100   150
/test5     32012* 20000   50000 7days       0   2000  3000
```

The following example shows quota information for the group `rsgusers`:

```
# quota -g rsgusers
Disk quotas for group rsgusers (gid 15):
Filesystem  blocks  quota  limit grace files quota limit  grace
/           118    200    300           2    20    40
/usr     13184* 10000   20000 7days       2    40    80
/test1    36136 100000  200000           124 10000 20000
/test2    44064 200000  400000           4   2000  4000
/test3     3587  30000   60000        628  3000  5000
/test4    51071 150000  300000           6   1050  1800
/test5    61044 100000  200000           3  10000 20000
```

#### 2.2.4.4 Verifying Quota Consistency

The `quotacheck` command verifies that actual block use and number of files are consistent with the established limits. It examines user and group files, builds a table of current disk usage, and compares this table with that stored in the disk quota file. If any inconsistencies are detected, AdvFS updates both the quota file(s) and the current system copy.

If you do not activate quotas automatically at system start-up (see Section 2.2.5), it is a good practice to run the `quotacheck` command when quotas are first activated (see Section 2.2.6). To ensure accuracy, run this command when there is no activity on the system.

The `quotacheck` command only checks filesets that have the `userquota` or `groupquota` option specified in the `/etc/fstab` file (see Section 2.2.2). By default both user and group quotas are checked, but you can specify either by selecting the `-u` option for user or the `-g` option for group.

The `quotacheck` command requires that filesets be mounted with quotas activated. Select the `-v` option (verbose) to display inconsistencies found and procedures performed during the checking process.

The following example shows a verbose check of the fileset `set_1` that displays no inconsistencies:

```
# quotacheck -v domain_1#set_1
*** Checking user and group quotas for domain_1#set_1 (/test1)
```

The following example checks all filesets that have quotas defined in the `/etc/fstab` file. In this example the `quotacheck` command fixes inconsistencies in `/usr`:

```
# quotacheck -va
*** Checking user and group quotas for /dev/rdisk/dsk0g (/usr)
*** Checking user and group quotas for domain_1#set_1 (/test1)
/usr: root    fixed: inodes 3057 -> 3022 blocks 100616 -> 123440
/usr: system fixed: inodes 2483 -> 2488 blocks 91721  -> 114568
/usr: adm     fixed: inodes 280  -> 240 blocks  487   ->    464
```

In this display, `inodes` is the number of files and `blocks` is the block usage.

#### 2.2.4.5 Summarizing Quotas by Fileset

The `repquota` command displays the actual disk usage and quotas for the specified filesets. To be included in the summary, the fileset must have a quota entry in the `/etc/fstab` file. By default both user and group quotas are reported, but you can specify either by using the `-u` option for user or the `-g` option for group.

For each user or group, the `repquota` command prints the current number of files, the amount of space used, and the quota limits established with the `edquota` command.

The following example summarizes quotas for a single fileset mounted on `/test1`:

```
# repquota -v /test1
*** Report for user quotas on /test1 (domain_1#set_1)
      Block limits      File limits
User      used  soft  hard  grace  used  soft  hard  grace
root  --   34088    0    0         123    0    0
user5  --    816 20000 30000         9   350   500
```

The following example displays user and group quota information for all filesets in `/etc/fstab` that have quotas defined. Note that this example contains both UFS and AdvFS files:

```
# repquota -va
*** Report for group quotas on /usr (/dev/disk/dsk0g)
      Block limits      File limits
Group      used  soft  hard  grace  used  soft  hard  grace
system  -- 114568    0    0         2488    0    0
daemon  --   144    0    0          1    0    0
uucp    --   801    0    0          8    0    0
mem     --  1096    0    0         10    0    0
bin     -- 108989    0    0        3219    0    0
mail    --   209    0    0          2    0    0
terminal --    56    0    0          2    0    0
adm     --   464    0    0        240    0    0
operator --   392    0    0          3    0    0
211     --  6937    0    0         33    0    0
*** Report for user quotas on /usr (/dev/disk/dsk0g)
      Block limits      File limits
User      used  soft  hard  grace  used  soft  hard  grace
root     -- 123440    0    0        3022    0    0
bi       -- 102534    0    0        2940    0    0
uucp     --   729    0    0          7    0    0
adm      --     1    0    0          1    0    0
user5    --    15   18   24          1    0    0
kraetsch --  6937    0    0         35    0    0
*** Report for group quotas on /test1 (domain_1#set_1)
      Block limits      File limits
Group      used  soft  hard  grace  used  soft  hard  grace
system  --  22816    0    0          50    0    0
daemon  --  12088    0    0          82    0    0
*** Report for user quotas on /test1 (domain_1#set_1)
      Block limits      File limits
User      used  soft  hard  grace  used  soft  hard  grace
root     --  34088    0    0         123    0    0
user5    --    816 20000 30000         9   350   500
```

```

*** Report for group quotas on /test3 (domain_2#set_1)
      Block limits      File limits
Group      used  soft  hard  grace  used  soft  hard  grace
system  --  1593      0      0      6      0      0
*** Report for user quotas on /test3 (domain_2#set_1)
      Block limits      File limits
User      used  soft  hard  grace  used  soft  hard  grace
root     --  1593      0      0      6      0      0

```

## 2.2.5 Activating Quotas at System Start-up

You can automatically start user and group quota enforcement during system initialization by modifying the `/etc/rc.config` file. Edit the `QUOTA_CONFIG` option to read:

```
QUOTA_CONFIG="yes"
```

This entry causes the `/sbin/init.d` quota script to run the `quotaon` and `quotacheck` commands.

Then, edit the `/etc/fstab` file entry to add `userquota` and `groupquota` to the mount point. Quota enforcement is activated for the mounted fileset the next time and every time you reboot.

For example, if you wanted to activate a quota on the fileset `WA` in the domain `expenses`, the `/etc/fstab` entry would be:

```
expenses#WA /WA advfs rw,userquota,groupquota 0 2
```

---

### Note

---

If you unmount a fileset when quota enforcement is active, you must explicitly reactivate quota enforcement with the `quotaon` command when you remount the fileset. This must be done even if there is a `QUOTA_CONFIG="yes"` entry.

---

## 2.2.6 Activating Quotas Manually

If your system is running, enter the `quotaon` command to activate new quotas for a mounted fileset. To establish new user or group quotas do the following:

1. Edit the `/etc/fstab` file entry for your fileset to add `userquota` and `groupquota` to the mount point.
2. Run the `edquota` command to enter the hard and soft limits and to enter the grace period.
3. Run the `quotaon` command to activate the quotas you have chosen.

If your system is set up to initialize quotas at system start-up (see Section 2.2.5), you do not need to run the `quotaon` command again unless you have unmounted your fileset. If your system does not include quota enforcement, you must run the `quotaon` command to start enforcement each time you reboot.

The following example activates quotas for the filesets for which quota values were set in the previous sections:

```
# quotaon -av
/test1: group quotas turned on
/test1: user quotas turned on
/test3: group quotas turned on
/test3: user quotas turned on
/test4: group quotas turned on
/test4: user quotas turned on
```

By default, both user and group quotas are affected by the `quotaon` and `quotaoff` commands. You can choose to activate quotas either for users (with the `-u` option) or for groups (with the `-g` option). You can also specify the filesets for which user or group quotas will be enforced (see Section 2.3.5 for information on fileset quotas).

## 2.2.7 Deactivating Quotas

You can turn off quota enforcement either temporarily or permanently. You can obtain file and disk space usage information regardless of whether you are enforcing quotas.

The `quotaoff` command turns off quota enforcement until the `quotaon` command is run again either manually or through system initialization that turns quotas on.

The `umount` command turns off quotas before it unmounts a fileset. If you remount the fileset, you must run the `quotaon` command to enforce user and group quotas for the fileset.

If you want to permanently turn quotas off for a user or group, use the `edquota` command to set quota limits to 0 (zero). To prevent quotas from ever being activated for a fileset, run the `quotaoff` command. Then, remove the `userquota` and `groupquota` entries for the fileset in the `/etc/fstab` file.

## 2.3 Fileset Quotas

The following sections describe fileset quota files and how quotas and grace periods are set for filesets from the command line. For information on setting fileset quotas from the AdvFS GUI, see Chapter 6.



## 2.3.1 Quota Files

AdvFS keeps fileset soft and hard limits in the structural information associated with the fileset. You do not have direct access to this file. It contains the same type of information that the user and group quota files contain: hard and soft limits for number of blocks and hard and soft limits for number of files. For a given fileset, the fileset grace period is the same as the group grace period.

## 2.3.2 Setting Quotas and Grace Periods

Fileset quotas limit the number of files or the amount of disk space a fileset can use. You can set both soft and hard limits. If fileset quotas are not imposed, any fileset has access to all of the available disk space in the domain. The fileset quotas are set with the `chfsets` command. If fileset quotas are set, they are activated whenever you mount the fileset.

The fileset grace period is the same as the group grace period (see Section 2.3.2.2). If you do not set a grace period, the grace period remains at the AdvFS default grace period of 7 days.

Use the `chfsets` command to define fileset quota values. You can set a soft limit for the number of files (`-F` option), a hard limit for the number of files (`-f` option), a soft limit for block usage (`-B` option), and a hard limit for block usage (`-b` option). The command displays both the old and new limits.

### 2.3.2.1 Fileset Quotas Example

The following example sets fileset quotas for the `set_1` fileset in `dmn_2`. Note that unlike the quota commands, the `showfsets` command displays block usage in 512-byte blocks. If you wish to display kilobyte values, use the `-k` option.

1. To display existing fileset quotas, use the `showfsets` command:

```
# showfsets dmn_2 set_1
set_1
  Id          : 2feff762.00034e3f.1.8001
  Clone is    : set_1_clone
  Files       :      7,  SLim=      0,  HLim=      0
  Blocks (512) :    118,  SLim=      0,  HLim=      0
  Quota Status : user=on  group=on
```

Here `SLim` is the soft limit and `HLim` is the hard limit for the number of files (`Files`) and the current block usage (`Blocks`).

2. Use the `chfsets` command to set the quotas. Note that the arguments for block usage for the `chfsets` command are in units of 1 kilobyte, not 512 bytes as shown by the `showfsets` command display.

```
# chfsets -F 10000 -f 20000 -B 250000 - b 500000 dmn_2 set_1
set_1
      Id           : 2feff762.00034e3f.1.8001
      File H Limit : 0 20000
      Block H Limit: 0 500000
      File S Limit : 0 10000
      Block S Limit: 0 250000
```

Here File H Limit is the hard limit for the number of files, Block H Limit is the hard limit for block usage, File S Limit is the soft limit for the number of files, and Block S Limit is the soft limit for block usage.

3. To verify the new fileset quotas, run the `showfsets` command again:

```
# showfsets dmn_2 set_1
set_1
      Id           : 2feff762.00034e3f.1.8001
      Clone is      : set_1_clone
      Files         :      7,  SLim= 10000,  HLim= 20000
      Blocks (512) :    118,  SLim= 500000,  HLim= 1000000
      Quota Status : user=on group=on
```

Note that the soft limit for the number of blocks is 500000, twice the number 250000 that was input with the `chfsets` command; similarly, the hard limit for the number of blocks appears to double from 500000 to 1000000. To avoid this confusion, enter the `showfsets` command with the `-k` option to display blocks in 1-kilobyte units.

### 2.3.2.2 Setting the Grace Period

The fileset grace period is the same as the group grace period and cannot be modified independently (see Section 2.2.2). Therefore, if you use the `edquota -gt` command to change the grace period for which a fileset can exceed its soft limits, you will also change the group grace period and vice versa. The default AdvFS grace period of 7 days remains in effect until you change it.

You can set only one grace period per fileset, but you can set different grace period values for block usage and number of files. The grace period applies to all users and all groups. If the grace period is reset, the new grace period for the fileset takes effect immediately unless the fileset has already exceeded its soft limits. In that case, the new grace period becomes effective once the fileset drops below the soft limit.

The following example sets the grace period for the filesets `test1`, `test3`, and `test4`:

1. Run the `edquota` command with the group grace period options. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file:

```
# edquota -gt
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/test1: block grace period: 7 days,file grace period:7 days
/test3: block grace period: 7 days,file grace period:7 days
/test4: block grace period: 7 days,file grace period:7 days
```

2. To change the fileset grace period for the number of blocks and for the number of files, edit the file to change the existing grace period:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/test1: block grace period: 7 days, file grace period:7 days
/test3: block grace period:12hours, file grace period:5 days
/test4: block grace period: 7 days, file grace period:7 days
```

3. Exit the editor, saving the changes.

Run the `showfsets` command with the `-q` option to see the time (if exceeded) and fileset usage and limits:

```
# showfsets -q test_domain
          Block (512) Limits  File Limits
Fileset  BF  used soft hard grace  used soft hard grace
fileset1 +- 1750 1500 2000 11:32    35  300  400
```

In this example, the plus sign (+) in the BF field means that the soft limit for block usage is exceeded. An asterisk (\*) indicates that the hard limit has been reached.

### 2.3.3 Setting Quotas for Multiple Filesets

You can set quota limits for multiple filesets by listing more than one fileset name when you run the `chfsets` command (see Section 2.3.2).

For example, to change the hard limits for the `data` and `data2` filesets in `test1_domain`, enter the names of both filesets after the `chfsets` command:

```
# chfsets -b 1000 -f 200 test1_domain data data2
data
      Id           : 2fdf591b.000855fa.2.8001
      File H Limit : 11  200
      Block H Limit: 121 1000
data2
      Id           : 2fdf591b.000855fa.3.8001
      File H Limit : 50  200
      Block H Limit: 200 1000
```

Setting fileset quotas automatically activates the quotas.

## 2.3.4 Verifying File and Disk Space Usage

To examine how system resources are being used, look at fileset activity. Table 2–3 contains commands that are useful for examining disk space and the file usage of filesets.

**Table 2–3: Fileset Disk Usage Commands**

Command	Description
<code>df</code>	Displays disk space used and available disk space for a fileset
<code>showfdmn</code>	Displays the attributes of a domain
<code>showfsets</code>	Displays the attributes of filesets in a domain
<code>vdf</code>	Displays disk space used and available disk space for a fileset or a domain

See the command reference pages for more information.

### 2.3.4.1 Displaying Used and Available Disk Space for Filesets

You can display the available disk space and the disk space used for a fileset with the `df` command (see Section 2.3.4.2 for domain information). The command calculates capacity using the lower (hard or soft) limit for the amount of space available:

- If a fileset quota has been set, the command displays the amount of space remaining until the quota limit is reached.
- When both soft and hard quota limits are set, the command calculates the disk space available using the lower limit.
- If there is less space in the domain than is allowed by the fileset quota, the command displays the actual space available in the domain.
- If fileset quotas have not been established, the command displays the available domain size; all unused space is available to each fileset.

The following example displays the amount of space available for `fileset_1`:

```
# df /fileset_1
Filesystem      512-blocks  Used Avail  Capacity  Mounted on
test_domain#fileset_1  1500  1750    0      117%  /fileset_1
```

Because the usage is over the limit, the capacity is determined by the actual space used (1750/1500) and appears as more than 100%. If usage is not over the limit, capacity is calculated as (used)/(used + available).

AdvFS calculates each fileset capacity independently. If the domain has multiple filesets, all unused space is available for each fileset unless the space is limited by fileset quotas. As a result, the total capacity of all filesets in the domain may appear to be greater than 100%. In the following example, the filesets `domain_1#test3` and `domain_1#test4` each can use all of the available disk space from the volumes in `domain_1`:

```
# df
  Filesystem      512-blocks    Used   Avail Capacity  Mounted on
domain_1#test3    2000000  390820   98864    80%    /test3
domain_1#test4    2000000  271580   98864    73%    /test4
```

#### 2.3.4.2 Displaying Used and Available Disk Space for Domains and Filesets

The `vdf` utility reformats output from the `showfdmn`, `showfsets`, and `df` commands to display information about the disk usage of AdvFS domains and filesets. It clarifies the relationship between a domain's disk usage and its fileset's disk usage.

The utility is subject to the following restrictions:

- All filesets in a domain must be mounted if you want to calculate disk usage of the domain.
- Disk space used by clone filesets is not calculated.
- The command does not produce valid results for filesets that are NFS mounted.

You can specify either a domain or a fileset name for the `vdf` command. If you specify only a fileset name, the output is the same as that of the `df` command. If you specify a domain, the utility also displays the number of disk blocks used for metadata. If you specify the `-1` option with either a domain or a fileset, both domain and fileset information will be displayed. The domain metadata displayed is the total metadata shared by all filesets in the domain.

The following example shows the summary information for the domain `usr_domain`:

```
# vdf usr_domain
Domain      512-blocks  Metadata    Used   Available  Capacity
usr_domain    65536      11219    47549     6768      89%
```

The following example examines the domain `test` that contains two filesets. Each has a quota limit of 60,000 blocks.

```
# vdf -l test
```

Domain	512-blocks	Metadata	Used	Available	Capacity
test	266240	5824	29128	231288	13%

Fileset	QuotaLimit	Used	Available	Capacity
testfs	60000	20800	39200	35%
testfs2	60000	8328	51672	14%

The following example shows the domain `test` that contains two filesets with no quota limits. In this case, the total space available to each fileset is the same as the domain total.

```
# vdf -l test
```

Domain	512-blocks	Metadata	Used	Available	Capacity
test	266240	5824	29128	231288	13%

Fileset	QuotaLimit	Used	Available	Capacity
testfs	-	20800	231288	35%
testfs2	-	8328	231288	14%

### 2.3.4.3 Displaying Fileset Attributes

The `showfsets` command with the `-q` option shows file usage, hard and soft limits, and grace period information for the filesets in the specified domain. It shows the block usage, the block usage limit, the number of files, and the file limit. The correct information will be displayed only if the fileset is mounted.

The following example shows fileset information for the domain `test_domain`:

```
# showfsets -q test_domain
```

		Block (512) Limits				File Limits			
Fileset	BF	used	soft	hard	grace	used	soft	hard	grace
fileset1	+-	1750	1500	2000	11:32	35	300	400	

In this example, the plus sign (+) in the `BF` field means that the soft limit for block usage is exceeded. An asterisk (\*) would indicate that the hard limit has been reached.

### 2.3.4.4 Displaying Domain Attributes

The `showfdmn` command is useful for obtaining domain statistics to make decisions about filesets and their quotas. The command shows the attributes of a domain and information about each volume in the domain. For single-volume or multivolume domains, the command shows the total volume size, the total number of free blocks, and the total percentage of volume space currently allocated.

### 2.3.5 Activating Quotas

Running the `chfsets` command automatically activates fileset quotas immediately (see Section 2.3.3). No further steps are needed. Fileset quotas are in effect whenever you mount the fileset.

### 2.3.6 Deactivating Quotas

You can turn off quota enforcement either temporarily or permanently by running the `chfsets` command with the hard and soft limits set to 0 (zero) to deactivate quotas on a fileset. You can obtain file and disk space usage information regardless of whether you are enforcing quotas.





---

## Backing Up and Restoring

AdvFS provides extended file system backup capabilities with the `vdump` and `vrestore` commands. The `rvdump` and `rvrestore` commands provide the same capabilities for remote storage devices. In addition, the AdvFS `clonefsset` utility can be used with the `vdump` command to back up online AdvFS filesets. Clone filesets provide a static snapshot of files for the backup.

The `vdump` command can be used to back up AdvFS filesets as well as UFS and other standard file systems. This means that you can have a single backup utility for your facility. The `dump` and `restore` commands function differently from the `vdump` and `vrestore` commands. The `dump` command works at the inode level so it can handle only UFS files. The `vdump` command works at the file level. It scans the directories and uses regular POSIX file system calls to access directories and files. See `vdump(8)` and `vrestore(8)` for more information.

In the discussion that follows, only the `vdump` and `vrestore` commands are mentioned, but the `rvdump` and `rvrestore` commands can be substituted for remote operation.

---

### Note

---

The tools you use to back up and to restore files must be compatible. For example, if you use the `vdump` or the `rvdump` command to back up a file system, you must use the `vrestore` or the `rvrestore` command to restore saved files. You cannot use the `vrestore` command to restore files backed up with the `dump` command.

---

You do not have to be root user to use the `vdump` and `vrestore` commands. However, the AdvFS quota files and the fileset quotas for the fileset can be saved and restored only when the root user initiates the command. You must have write permission for the directory to which you want to restore files.

---

### Note

---

The `vrestore` command for operating system versions earlier than Version 4.0 will not properly restore information saved by the `vdump` command for Version 4.0 or later. The `vdump` and `vrestore` dump file formats are compatible in Version 4 and Version 5.

---

## 3.1 Backing Up Data

The `vdump` command creates a list of fixed-size blocks, called a *saveset*, as it copies all files that are new or have changed after a certain date to the default storage device or the device that you specify.

A *saveset* can span multiple tapes or a tape can contain multiple *savesets*. *Savesets* on tapes are delimited by file marks that are written when the *saveset* is closed by the `vdump` command.

To enter a `vdump` command:

```
vdump options mount_point
```

For example, to dump the filesets mounted at `/psm` to tape:

```
# vdump -0 -f /dev/tape/tape0 /psm
```

### 3.1.1 Unique Features of the `vdump` Command

The `vdump` command has a number of functions that the UFS `dump` command does not have. You can:

- Save mounted filesets.
- Choose a subdirectory that you want to back up. You do not need to dump an entire fileset.
- Compress files to minimize the *saveset* size.
- Specify the number of in-memory buffers. You can maximize throughput by choosing a number compatible with your storage device.
- Display the current `vdump` version number.
- Display help information during the dump process.
- Limit your display to error messages. You do not need to display warning messages.
- Display the names of files as they are backed up.

- Configure output with an error-protection system that will allow you to recover data even if there is a read error when you restore.
- Handles AdvFS and UFS sparse files without zero fills.

### 3.1.2 Dumping to Tape

You can place multiple savesets on one tape with the `vdump` command. Set the `-N` option to specify no rewind or specify a no-rewind device such as `/dev/ntape/tape0`. This ensures that the tape does not rewind when the dump finishes.

If a saveset requires more than one tape to complete, you will be prompted to mount another.

Do not combine the output from the `dump` and `vdump` utilities on the same tape. If the `vrestore` command is used to recover files from a tape created by the `dump` utility, the results are unpredictable and can result in data loss.

### 3.1.3 Dumping Subdirectories

You can selectively back up individual subdirectories of a filesset by specifying the subdirectory with the `-D` option of the `vdump` command. Without the `-D` option, if you specify a subdirectory instead of a filesset on the command line, the `vdump` command backs up the entire filesset that contains the named subdirectory. If you specify the `-D` option, backup is always run at level 0.

### 3.1.4 Dumping to Filessets

When the backup saveset device is the character dash (`-`), the `vdump` command writes to standard output. Thus, the `vdump` and `vrestore` commands can be used in a pipeline expression to copy one filesset to another. The following are typical commands; they are equivalent:

```
#vdump -0f - /usr | vrestore -xf - -D /mnt
```

```
#vdump -0 -f - /usr | (cd /mnt; vrestore -x -f -)
```

The `rvdump` and `rvrestore` commands are unable to use the character `-`. The output device must be specified.

### 3.1.5 Dumping to Block 0

If you attempt to dump to the first block of a disk partition that contains a valid disk label, the device driver will not overwrite the disk label. You will get an error message only if you are using a character or raw device; block special devices will not return an error.

If you want to dump to block 0, you must first clear the disk label. If you do not, the `vdump` command may appear to contain valid savesets, but when the `vrestore` command attempts to interpret the disk label as part of the saveset, it will return the message:

```
vrestore: unable to use save-set; invalid or corrupt format
```

This is also true for the `rvdump` and `rvrestore` commands.

To correctly dump to the first disk block, begin by clearing the disk label with the `disklabel` command and the `-z` flag. See `vdump(8)` and `vrestore(8)` for more information.

### 3.1.6 Compressing Savesets

You can compress savesets as they are backed up. This reduces the amount of storage required for the backup and allows the dump to run faster on slow devices because less data is written. Use the `-c` option with the `vdump` command to request compression. You cannot specify the compression ratio; it is determined by the contents of the dump.

---

#### Note

---

If you are using a tape drive that automatically does hardware compression, using the `vdump` command with compression may result in a larger saveset than expected. Sometimes, due to compression algorithms, already compressed data gets expanded when an attempt is made to do more compression.

---

### 3.1.7 Dumping with Error Protection

You can use the `-x` option with the `vdump` command to place checksum blocks on your tape so that the `vrestore` command can recover damaged blocks. The `vdump` command creates these blocks every `n` number of blocks you specify. The valid range of `n` is 2 to 32; the default is 8. If the `vrestore` command detects a read error in a block, it uses the other blocks and the checksum block to recreate the bad block.

Dumping with error protection requires saving one extra block for every `n` blocks. It can correct only one block in each series of `n` blocks when the blocks are restored. This means there is a trade-off:

- If you believe tapes are error prone or you require extremely accurate backups and you have many tapes available for backup, set the value of `-x` to 2. This will permit error correction of one bad block for every two blocks saved. It will require 50% more tape because after every two dump blocks, a checksum block will be written.

- If you believe that tapes are generally reliable but you want to be able to correct a rare bad block, set the value of `-x` to 32. This will require 3% more tape because an extra block will be added for every 32 blocks written. You could then recover information from any one bad block in the group of 32 dump blocks.

### 3.1.8 Backup Level

You can specify the level of incremental backup in the `vdump` command. A value of 0 specifies complete fileset backup. A higher number specifies a less complete backup. See `vdump(8)` for more information.

The `vdump` command operates by checking the file modification date. This may cause problems on a subsequent incremental backup because the file modification date does not change if you rename or move a file and don't modify the data. Thus, if you back up your files then move or rename them, change the modification date using the `touch` command:

```
touch file_name
```

### 3.1.9 Listing Saved Files

You can check your saveset and make sure you have backed up the files you intended. After your backup is complete, run the `vrstore` command with the `-t` option to display the files you have saved. This will not initiate the restore procedure.

### 3.1.10 Dumping and Restoring Files Remotely

The `rvdump` command backs up files from a single mounted fileset or a clone fileset to a remote storage device. You must be able to execute the `rsh` command on the remote node to which you are dumping. See `rsh(8)` for server and client access rules.

The `rvdump` command has the same options as the `vdump` command, but you must specify the node name for the device that you are backing your files to. The following example dumps a fileset `sar` to a tape on node `rachem`:

```
# rvdump -0f rachem:/dev/tape/tape0 /sar
```

To restore the fileset from the remote tape drive, enter:

```
# rvrestore -xf rachem:/dev/tape/tape0 -D /sar
```

## 3.2 Cloning for Online Backup

A clone fileset is a read-only snapshot of the data in an existing fileset. If you have the optional AdvFS Utilities, you can create a clone. You must be the root user to clone the root fileset.

A clone does not contain all the data at the time it is created. Rather, it is an outline of the data structure. Data in the original fileset that remains unchanged over the life of the clone is never saved in the clone fileset. When you modify the data in your original files, AdvFS saves the data that existed in the original, page by page, into the clone.

You can create a clone fileset for any AdvFS fileset with the command-line interface (see Section 1.5.10) or with the AdvFS GUI (see Section 6.4.3). You cannot clone UFS file systems. Only one clone can exist per fileset.

A clone is useful if the files in your system are changing during the time you wish to do your backup. It reflects the state of your system at the moment it was created and backing it up does not interfere with current processing.

---

### Note

---

After you have finished your backup, delete the clone. Clones of active filesets will continue to grow as the files are changed.

---

The following example backs up the `pssm` fileset on line by creating the `pssm_clone` fileset and backing it up to the default device. The domain in this example is `domain1`.

```
# clonefset domain1 pssm pssm_clone
# mkdir /pssm_clone
# mount -t advfs domain1#pssm_clone /pssm_clone
# vdump -0 -u -C /pssm_clone
```

To remove the `pssm_clone` fileset, enter:

```
# umount /pssm_clone
# rmfset domain1#pssm_clone
```

## 3.3 Cloning to Back Up Databases

If your database has an online backup utility, use it to backup the database. If it does not, you can back up databases with database down time limited to the short time it takes to create the clone fileset. Backing up a database with a clone fileset is the same as backing up any other fileset. You get the same benefits (see Section 3.2).

To back up a database with a clone:

1. Shut down the database so that all database buffers are flushed and the fileset has a complete, consistent copy of the database files.
2. Clone the fileset and mount the clone.
3. Reactivate the database.
4. When you want to back up the clone fileset, run a backup procedure.
5. Unmount and delete the clone.

---

**Caution**

---

Do not use anything except the database's own utilities to back up an active database. You can use the `vdump` and `vrestore` commands on a clone fileset that contains the database.

---

If your database has files spread over multiple AdvFS filesets, it is a good idea to create a clone on each of the filesets at the same time. This ensures a consistent back up of all the database information.

## 3.4 Restoring Data

The `vrestore` command restores files by processing the blocks from a `saveset` created with the `vdump` command. The `vrestore` command will not work on a `saveset` created by the UFS `dump` command.

You do not have to be root user to run the `vrestore` command, but you must have write privilege for the directory you will restore to. Only the root user can restore quota files and fileset quotas. See `vrestore(8)` for details.

### 3.4.1 Unique Features of the `vrestore` Command

The `vrestore` command performs a number of activities that the UFS `restore` command does not. You can:

- Display the current `vrestore` version number.
- Display the source directory path.
- List the `saveset` structure.
- Display error messages only. Information messages will not be shown.
- Specify how the `vrestore` command should proceed if it encounters a file that already exists. You can choose whether the command will always overwrite an existing file, never overwrite an existing file, or query you for each event.

### 3.4.2 Restoring Files

The `vrestore` command allows you to select specific files and directories to be restored. It can restore data from a file, a pipe, magnetic tapes, or disks.

Use the same version of the `vdump` and `vrestore` utilities. If your version of the `vrestore` utility is unable to read the format of your saveset, you will get an error message.

Before you restore files, you can check if the saveset you are accessing contains the information you wish to recover. You can list the names and sizes of all files in the saveset by running the `vrestore` command using the `-t` option. The restore operation will not be performed. You can also display the files and directories saved by running the `vrestore` command with the `-i` option. This interactive option allows you to select individual files or directories to restore from a list.

Restoring data from a clone fileset is the same as restoring data from any other fileset.

Start with the full backup if you are restoring an entire fileset. Then restore later incremental backups on top of this to retrieve files that have changed since the full backup was created. Files that were deleted after the full backup was performed are restored. It is necessary to delete these files manually.

### 3.4.3 Restoring Quotas

AdvFS user and group quota files can be restored either to an AdvFS fileset or to a UFS file system. If AdvFS quota files are to be restored to a UFS file system, quotas must be activated on the UFS file system. AdvFS fileset quotas cannot be restored to a UFS file system because there is no UFS analog to AdvFS fileset quotas. You must be root user to restore quotas.

### 3.4.4 Restoring Selected Savesets

To restore to the current working directory from a tape containing multiple savesets, use the `mt` command with the `fsf n` (forward space n savesets or files) option to locate the saveset you want to restore. Then use the `vrestore` command.

The following example selects and restores the fourth saveset on a tape:

```
# mt fsf 3
# vrestore -xf /dev/ntape/tape0
```

The `vrestore` command can also selectively restore files from your saveset with the `-x` option followed by the file names. You can specify a destination path other than the current directory for the restored files.



The following example restores the file named `data_file` from the `/mnt/fdump` save-set. It is restored to the `/mnt` directory.

```
# vrestore -f /mnt/fdump -D /mnt -x data_file
vrestore: Date of the vdump save-set: Thu Jun 15 15:27:36 2000
```

### 3.5 AdvFS and NetWorker

NetWorker for Tru64 UNIX provides scheduled, online, automated backup. Use NetWorker with AdvFS as a comprehensive backup solution. NetWorker can automatically back up multiple servers in a heterogeneous environment. It has a graphical interface and several scheduling options.

If filesets will be accessed by users during the backup process, use the AdvFS `clonefsset` utility to clone all filesets for backup and mount the clone filesets. (You can create a script to accomplish this task.) Then, set up NetWorker to automatically back up the clone filesets on a convenient schedule.



---

## Optimizing Performance

AdvFS provides a number of ways to configure and tune your file system. Some of the tuning functions are available through the AdvFS GUI (see Chapter 6). The System Configuration and Tuning manual provides detailed information.

### 4.1 AdvFS Organization

You will obtain the best performance if you carefully plan your AdvFS configuration. You can control how you configure your domains and how you allocate disks. You can turn on direct I/O to speed data transfer. You can make choices about transaction logging and file structure.

#### 4.1.1 Configuring Domains and Filesets

There is a trade-off in using one large domain instead of several smaller ones. Because each domain has one transaction log, creating a single large domain decreases maintenance complexity at the cost of putting a greater load on the log, which may become a bottleneck (see Section 1.4.1).

Domains that were created on operating systems prior to Version 5.0 do not have the structure necessary to provide large quota values and better performance for directories containing thousands of files. If either of these new features is important to you, update your domains (see Section 1.4.3).

Multiple filesets in a domain are generally more efficient than a single large one (see Section 1.5.1).

See System Configuration and Tuning for more detailed information about allocating domains and filesets effectively.

#### 4.1.2 Configuring Volumes

If you have AdvFS Utilities, you can add multiple volumes to an AdvFS domain. This may improve performance because I/O processes can run in parallel. However, without LSM disk mirroring, it is inadvisable to add more than eight volumes. If you lose a volume, the entire domain becomes inaccessible. The risk of losing a volume, and thus losing access to your domain, increases as the number of volumes increases.

In many cases, there is a significant performance advantage to dividing disks on different SCSI busses. See System Configuration and Tuning for more detailed information.

### 4.1.3 Improving Transaction Log Performance

Each domain has a transaction log that keeps track of fileset activity for all filesets in the domain. This requires a high volume of read/write activity to the log file. If the log resides on a congested disk or bus, or if the domain contains many filesets, system performance can degrade. You can shift the balance of I/O activity so that the log activity does not use up the bandwidth of the device.

Monitor performance of the volume with the SysMan “View Input/Output (I/O) Statistics” or with the `iostat` utility. If you have AdvFS Utilities, do one of the following if the volume containing the log appears overloaded:

- Divide the domain into several smaller domains. Because each domain has its own transaction log, each log will then handle transactions for fewer filesets.
- Move the transaction log to a faster or less congested volume.
- Isolate the transaction log on its own volume.

Moving the transaction log may also be useful when you are using LSM and wish to increase reliability by placing your transaction log on a volume that is mirrored.

To move the transaction log to another volume:

1. Use the `showfdmn` command to determine the location of the log. The letter `L` after the volume number indicates the volume on which the log resides.
2. Use the `switchlog` command to move the log to another volume.

For example, to move the transaction log for the domain `region1`:

```
# showfdmn region1
      Id              Date Created      LogPgs Version Domain Name
31bf51ba.0001be10 Wed Feb  9 16:24 2000  512          4 region1

Vol  512-Blks      Free % Used Cmode Rblks Wblks Vol Name
  1L  1787904   885168   52%   on   128   128 /dev/disk/dsk0g
    2   1790096 1403872   22%   on   128   128 /dev/disk/dsk0h
-----
      3578000 2259040   37%
```

```
# switchlog region1 2
# showfdmn region1
      Id          Date Created      LogPgs Version Domain Name
31bf51ba.0001be10 Wed Feb  9 16:24 2000  512          4 region1

Vol  512-Blks      Free % Used Cmode Rblks Wblks Vol Name
  1   1787904   885168    52%   on   128   128 /dev/disk/dsk0g
 2L   1790096  1395680    22%   on   128   128 /dev/disk/dsk0h
-----
      3578000  2250848    37%
```

Isolating the transaction log will allow all log I/O to be separate from other domain reads and writes. As there will be no other activity on this volume, the log I/O will not be slowed down and will not slow down other domain I/O.

To isolate the transaction log on its own volume:

1. Add a small partition (volume) to the domain for which you are going to isolate the log.

Remember that the I/O load of other partition(s) on the device will affect the performance of the entire disk including the log partition.

If the remaining partitions are allocated to other domains, there may be more than one transaction log on the same device. This may not be a problem on a solid state disk but may negate the value of isolating the log on slower devices.

2. Use the `switchlog` command to move the log to another volume.
3. Use the `showfdmn` command to determine the number of free blocks on the volume with the log.
4. With the `showfdmn` information, use the `dd` command to build a dummy file of the right size.
5. Migrate the dummy file to the volume that contains the log. This fills the volume completely and leaves no space for other files. Because you never access this file, only the transaction log file will be active on the volume.

For example, to isolate the transaction log for the domain `sales`:

```
# addvol /dev/disk/dsk9a sales
# switchlog sales 2
```

```
# showfdmn sales
```

Id	Date Created	LogPgs	Version	Domain Name
312387a9.000b049f	Thu Mar 16 14:24 2000	512	4	sales

Vol	512-Blks	Free	% Used	Cmode	Rblks	Wblks	Vol Name
1	2050860	1908016	7%	on	128	128	/dev/disk/dsk10c
2L	131072	122752	6%	on	128	128	/dev/disk/dsk9a
-----							
	2181932	2030768	7%				

Allocate all the free blocks on the volume containing the log to a dummy file, /adv1/foo, then move the data to the log volume :

```
# dd if=/dev/zero of=/adv1/foo count=122752
122752+0 records in
122752+0 records out
# migrate -d 2 /adv1/foo
```

#### 4.1.4 Improving Data Consistency

The method you choose to write data can affect what is saved if a machine fails. Following are several ways of writing to a file:

- Asynchronous I/O (default)

Write requests, by default, are cached; that is, data is written to the buffer cache, not immediately to disk. This method generally gives the highest throughput, in part because multiple writes to the same page can be combined into one physical write to disk. This not only decreases disk traffic, but it increases the concurrent access of common data by multiple threads and processes. In addition, delaying the write to disk increases the likelihood that a page may be combined with contiguous pages into a single, larger physical write, saving seek time and delays caused by rotational latency.

If a crash occurs, the next time a fileset in the domain is mounted, the completed log transactions are replayed to disk and incomplete transactions are backed out so that the original data on disk is restored. These log transactions, by default, save only metadata, not the data written to the file. This means that file sizes and locations on disk will be consistent but, depending on when the crash occurred, the data from recent writes may be out of date. This is a trade-off for the increased throughput gained using this method.

- Asynchronous atomic-write data logging I/O

This method is similar to asynchronous I/O except that the data written to the buffer cache is also written to the log for each write request. This is done in 8-kilobyte increments. Eventually the data is also written to the file, meaning that the data will be written to disk twice: once to the log and then to the file. The extra write of the data to the log may degrade throughput compared with using asynchronous I/O.

If a crash occurs, the data is recovered from the log when the fileset is remounted. As is done in asynchronous I/O, all completed log transactions are replayed and incomplete transactions are backed out. Unlike asynchronous I/O, however, the user's data has been written to the log, so both the metadata and the data intended for the file can be restored. This guarantees that each 8-kilobyte increment of a write is either completely written to disk or not written. Because only completed write requests are processed, obsolete, possibly sensitive data located where the system was about to write at the time of the crash can never be accessed. Out-of-order disk writes, which might cause inconsistencies in the event of a crash, can never occur.

To turn atomic-write data logging I/O on and off, use the `fcntl()` function or enter the `chfile` command with the `-L` option:

```
chfile -L on file_name
```

```
chfile -L off file_name
```

If a file has a frag, atomic-write data logging cannot be activated. To activate data logging on a file that has a frag, append enough bytes to the file to bring it up to the next 8-kilobyte boundary. For example, if `fileb` had 6803 bytes, it would be stored in one 7-kilobyte frag. To activate data logging, you would need to add 1389 bytes so the file would terminate on an 8-kilobyte boundary:

```
dd if=/dev/zero of=fileb bs=1 seek=6803 count=1389 \  
conv=notrunc
```

Files that use atomic-write data logging cannot be memory mapped through the `mmap` system call. See Section 5.2 for information on conflicting file usage.

- Synchronous I/O

Synchronous I/O is similar to asynchronous I/O, but the data is written both to the cache and to the disk before the write request returns to the calling application. This means that if a write was successful, the data is guaranteed to be correct. The penalty for this is reduced throughput because the write will not return until after the I/O has completed. In addition, since the application, not the file system, determines when the data needs to be flushed to disk, the likelihood of consolidating I/Os may be reduced if synchronous write requests are small.

To turn synchronous I/O off and on, use the `O_SYNC` or `O_DSYNC` flag to the `open()` system call (see the Programmer's Guide). To force all applications to synchronous I/O even if files are not opened in that mode, enter the `chfile` command with the `-l` option:

```
chfile -l on file_name
```

```
chfile -l off file_name
```

- Synchronous atomic-write data logging I/O

If you have activated atomic-write data logging on a file, you can open the file for synchronous I/O with the `O_SYNC` or `O_DSYNC` flag to the `open()` system call (see the Programmer's Guide).

The `fcntl()` function can be used to turn synchronous writes and atomic-write data logging on and off. See `fcntl(2)` and the Programmer's Guide for more information.

#### 4.1.5 Improving Data Transfer Rate with Direct I/O

You can use direct I/O mode to synchronously read and write data from a file without copying the data into a cache (the normal AdvFS process). That is, when direct I/O is enabled for a file, read and write requests on it are executed to and from disk storage through direct memory access (similar to raw I/O), bypassing AdvFS caching. This may improve the speed of the I/O process for applications that access data only once.

Although direct I/O will handle I/O requests of any byte size, the best performance will occur when the requested transfer size is aligned on a disk sector boundary and the transfer size is an even multiple of the underlying sector size (currently 512 bytes).

Direct I/O is particularly suited for files that are used exclusively by a database. However, if an application tends to access data multiple times, direct I/O can adversely impact performance because caching will not occur. As soon as you specify direct I/O, it takes precedence and any data already in the buffer cache for that file will automatically be flushed to disk.

To open a file for direct I/O, use the `open()` function and specify the `O_DIRECTIO` flag. For example, for `file_x` enter:

```
open (file_x, O_DIRECTIO|O_RDWR, 0644)
```

If the file is already open for direct I/O or is in cached mode, the new mode will be direct I/O and will remain so until the last close of the file. Note that direct I/O, atomic-write data logging, and `mmap`ing are mutually exclusive modes. Therefore, if the file is already open for atomic-write data logging or is `mmap`ped, then calling the `open` function to initiate direct I/O will fail.



The `fcntl()` function can be used to determine whether the file is open in cached or in direct I/O mode. See `fcntl(2)` and `open(2)` or the Programmer's Guide for more information.

## 4.2 Monitoring Performance

There are a number of ways to gather performance information:

- The `iostat` utility reports I/O statistics for terminals, disks, and the CPU. It displays the number of transfers per second (tps) and bytes per second in kilobytes (bps). From this you can determine where I/O bottlenecks are occurring. That is, if one device shows sustained high throughput, this device is being utilized more than others. Then you can decide what action might increase throughput: moving files, obtaining faster volumes, striping files, and so on. You can view I/O statistics with the SysMan "View Input/Output (I/O) Statistics" or from the command line (see `iostat(1)`).
- The `advfsstat` utility displays detailed information about the activity of filesets and domains over time. You can examine, for example, the activity of the buffer cache, volume reads/writes, and the BMT record. See `advfsstat(8)` for more information.
- Collect for Tru64 UNIX gathers and displays information for subsystems such as memory, disk, tape, network or file systems. Collect runs on all supported releases of Tru64 UNIX. For more information, contact `collect_support@compaq.com`

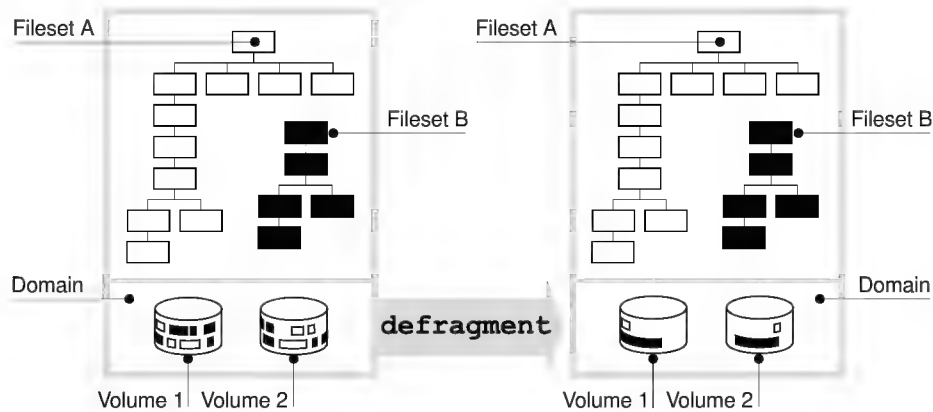
## 4.3 Tuning AdvFS

There are a number of things you can do to operate AdvFS more efficiently. You can defragment a domain, balance a multivolume domain to even the storage distribution, stripe files across disks to improve read/write performance, and migrate files to faster volumes. You can change caching attributes, I/O transfer parameters, and other AdvFS attributes. Detailed information about tuning is available in System Configuration and Tuning.

### 4.3.1 Defragmenting a Domain

AdvFS attempts to store file data in contiguous blocks on a disk. This collection of contiguous blocks is called a file extent. If all data in a file is stored in contiguous blocks, that file has one file extent. However, as files grow, contiguous blocks on the disk may not be available to accommodate the new data, and the system will spread the file over discontinuous blocks. As a result, the file is fragmented on the disk and consists of multiple file extents. File fragmentation degrades the read/write performance because many disk addresses must be examined to access a file.

**Figure 4–1: Defragmenting a Domain**



The `defragment` utility reduces the amount of file fragmentation in a domain by attempting to make the files more contiguous. Defragmentation, as illustrated in Figure 4–1, is an iterative, two-step process that operates on the domain:

1. Files are moved out of a region to create an area with contiguous, unallocated space.
2. Fragmented files are written into a region that has more contiguous space so they are less fragmented.

In addition to making files contiguous so that the number of file extents is reduced, defragmenting a domain often makes the free space on a disk more contiguous so files that are created later will also be less fragmented.

Files may be moved to other volumes in the defragmentation of a multivolume domain. You cannot control the placement of files as defragmentation occurs, but you can identify where a file is stored with the `showfile` command. If you want to move a file, use the `migrate` command (see Section 4.3.3).

You can improve the efficiency of the `defragment` process by deleting any unneeded files in the domain before running the `defragment` utility. Aborting the defragment process does not damage the file system. Files that have been defragmented remain in their new locations.

It is difficult to specify the load that defragmenting will place on a system. The time it takes to defragment a domain depends on:

- The size of the volume(s).
- The amount of free space available.
- The activity of the system.

- The configuration of your domain. A domain consisting of several small volumes is faster to defragment than one consisting of one large volume.

To defragment a domain, use the SysMan “Defragment an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `defragment` command:

```
defragment domain_name
```

The following restrictions apply to running the `defragment` command:

- You must have root user privileges.
- All filesets in the domain must be mounted. If you try to defragment an active domain that includes unmounted filesets, you will get an error message.
- A minimum free space of 1% of the total space or 5 megabytes per volume (whichever is less) must be available.
- The `defragment` utility cannot be run while the `addvol`, `rmvol`, `balance`, or `rmfset` command is running in the same domain.

See `defragment(8)` for more information.

#### 4.3.1.1 Choosing to Defragment

Run `defragment` on your domain when administratively necessary, and then only when file system activity is low. Run the `balance` utility before you run `defragment`. This will speed up the defragmentation process.

If your file system has been untouched for a month or two, that is, if you do not run full periodic backups nor regularly reference your whole file system, it is a good idea, before you run `defragment`, to run the `verify` utility. Run `verify` when there is low file system activity.

It is not efficient to balance your files after you defragment because this may undo some of the defragmentation and free space consolidation.

How fragmented you should let your file system become before running the utility depends on the size of the files and the number of extents. This is largely application dependent, so monitor the number of extents (see `defragment(8)`) to see if elevated extent counts correlate with decreased application performance. In many cases, even a large, fairly fragmented file will show no noticeable decrease in performance because of fragmentation.

It is not necessary to run `defragment`:

- If most of your files are less than 8 kilobytes.
- On write-only domains.

- On any system that is not experiencing performance-related problems because of excessive file fragmentation.
- On mail servers.

To determine the amount of file fragmentation that exists in a domain, run `defragment` with the `-v` and `-n` options. This will show how fragmented your domain is without altering the domain.

#### 4.3.1.2 Defragment Example

You can defragment a file domain from the SysMan “Defragment an AdvFS Domain,” from the command line, or from the AdvFS GUI (see Section 6.4.5.2).

From the command line:

1. To decide if defragmenting is necessary, run the `defragment` utility with the `-v` and `-n` options to look at the file defragmentation in the domain without starting the process.  
or  
Use the `showfile` command to check the file extents for a particular file in the domain.
2. Run the `defragment` utility specifying how long you want the process to continue.

The following example looks at the fragmentation of the `accounts_domain` file domain and at the number of extents in the `orig_file_1` file. It then defragments the domain for a maximum of 15 minutes. Verbose mode is requested to display the fragmentation data at the beginning of each pass through the domain and at the end of the defragmentation process.

```
# defragment -vn accounts_domain
defragment: Gathering data for 'accounts_domain'
Current domain data:
  Extents:                263675
  Files w/ extents:       152693
  Avg exts per file w/exts: 1.73
  Aggregate I/O perf:     70%
  Free space fragments:   85574
                        <100K  <1M   <10M  >10M
  Free space:    34%   45%   19%   2%
  Fragments:    76197 8930   440    7
# showfile -x orig_file_1
  Id Vol PgSz Pages XtntType Segs SegSz  I/O Perf  File
6.8002 2 16 71 simple ** ** async 82% orig_file_1
  extentMap: 1
            pageOff    pageCnt    vol    volBlock    blockCnt
                0         5        2      40720         80
```

5	12	2	41856	192
17	16	2	40992	256
33	7	2	42048	112
40	12	2	41360	192
52	15	2	42160	240
67	4	2	41792	64

extentCnt: 7

```
# defragment -v -t 15 accounts_domain
defragment: Defragmenting domain 'accounts_domain'
```

Pass 1;

Volume 2: area at block 144 ( 130800 blocks): 0% full  
Volume 1: area at block 468064 ( 539008 blocks): 49% full  
Domain data as of the start of this pass:

Extents:	7717			
Files w/extents:	6436			
Avg exts per file w/exts:	1.20			
Aggregate I/O perf:	78%			
Free space fragments:	904			
	<100K	<1M	<10M	>10M
Free space:	4%	5%	12%	79%
Fragments:	825	60	13	6

Pass 2;

Volume 1: area at block 924288 ( 547504 blocks): 69% full  
Volume 2: area at block 144 ( 130800 blocks): 0% full  
Domain data as of the start of this pass:

Extents:	6507			
Files w/extents:	6436			
Avg exts per file w/exts:	1.01			
Aggregate I/O perf:	86%			
Free space fragments:	1752			
	<100K	<1M	<10M	>10M
Free space:	8%	13%	11%	67%
Fragments:	1574	157	15	6

Pass 3;

Domain data as of the start of this pass:

Extents:	6522			
Files w/extents:	6436			
Avg exts per file w/exts:	1.01			
Aggregate I/O perf:	99%			
Free space fragments:	710			
	<100K	<1M	<10M	>10M
Free space:	3%	11%	21%	65%
Fragments:	546	126	32	6

Defragment: Defragmented domain 'accounts\_domain'

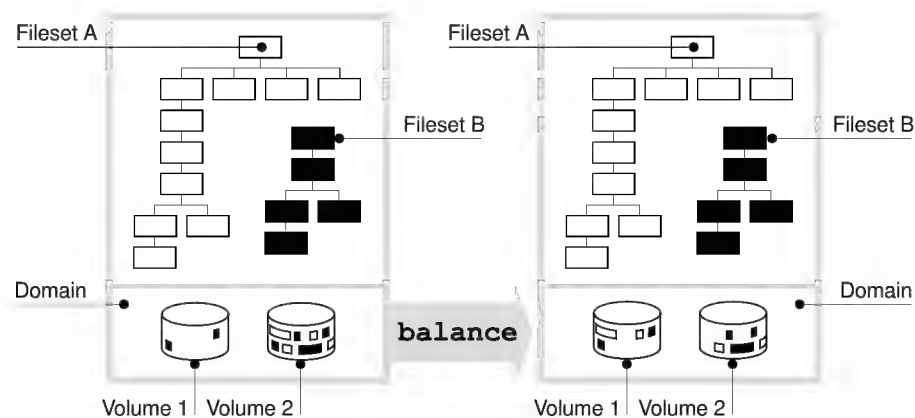
Information displayed before each pass and at the conclusion of the defragmentation process indicates the amount of improvement made to the

domain. A decrease in the `Extents` and `Avg exts per file w/extents` values indicates a reduction in file fragmentation. An increase in the `Aggregate I/O perf` value indicates improvement in the overall efficiency of file-extent allocation.

### 4.3.2 Balancing a Multivolume Domain

The `balance` utility distributes the percentage of used space evenly between volumes in a multivolume domain created with the optional AdvFS Utilities. This improves performance and evens the distribution of future file allocations.

**Figure 4–2: Balancing a Domain**



Files are moved from one volume to another, as illustrated in Figure 4–2, until the percentage of used space on each volume in the domain is as equal as possible. Because the `balance` utility does not generally split files, domains with very large files may not balance as evenly as domains with smaller files.

To redistribute files across volumes use the SysMan “Manage an AdvFS Domain,” the AdvFS GUI (see Chapter 6) or, from the command line, enter the `balance` command:

```
balance domain_name
```

If you interrupt the balance process, all relocated files remain at their new locations. The rest of the files remain in their original locations.

The following restrictions apply to running the `balance` utility:

- You must have root user privileges.
- All filesets in the domain must be mounted. If you try to balance an active domain that includes unmounted filesets, you will get an error message.

- A minimum free space of 1% of the total space or 5 megabytes per volume (whichever is less) must be available.
- The `balance` utility cannot run while the `addvol`, `rmvol`, `defragment`, or `rmfset` command is running in the same domain.

See `balance(8)` for more information.

#### 4.3.2.1 Choosing to Balance

Use the `showfdmn` command to display domain information. From the `% used` field you can determine if the files are evenly distributed.

Use the `balance` utility to even file distribution after you have added a volume with the `addvol` command or removed a volume with the `rmvol` command (if there are multiple volumes remaining).

#### 4.3.2.2 Balance Example

In the following example, the multivolume domain `usr_domain` is not balanced. Volume 1 has 63% used space while volume 2, a smaller volume, has 0% used space (it has just been added). After balancing, both volumes have approximately the same percentage of space used.

```
# showfdmn usr_domain
      Id          Date Created      LogPgs Version Domain Name
3437d34d.000ca710 Mon Apr 3 10:50:05 2000 512          4 usr_domain

Vol  512-Blks    Free % Used   Cmode Rblks  Wblks  Vol Name
1L   1488716  549232    63%    on   128   128  /dev/disk/dsk0g
2    262144  262000     0%    on   128   128  /dev/disk/dsk4a
-----
      1750860  811232    54%
```

```
# balance usr_domain
balance: Balancing domain 'usr_domain'
balance: Balanced domain 'usr_domain'
# showfdmn usr_domain
      Id          Date Created      LogPgs Version Domain Name
3437d34d.000ca710 Mon Apr 3 10:50:05 2000 512          4 usr_domain

Vol  512-Blks    Free % Used   Cmode Rblks  Wblks  Vol Name
1L   1488716  689152    54%    on   128   128  /dev/disk/dsk0g
2    262144  122064    53%    on   128   128  /dev/disk/dsk4a
-----
      1750860  811216    54%
```

### 4.3.3 Moving Files to Different Volumes

If you suspect that a fileset or domain is straining system resources, run the `iostat` utility either from the SysMan “View Input/Output (I/O) Statistics,” or from the command line (see `iostat(1)`). If the filesets or domains are located on devices that appear to be a bottleneck, you can migrate files or pages of files to equalize the load. If a high-performance device is available, you can move an I/O-intensive file to the more efficient volume.

If you do not have AdvFS Utilities, create a backup to move files using the dump and restore procedure. It is a good idea to mount the filesets you are moving as read only or to keep users from accessing the filesets at the time you are moving your files.

To move files:

1. Make a new domain on the new device. It must have a temporary new name.
2. For each fileset in the old domain, create a fileset with the same name in the corresponding new domain.
3. Create a temporary mount point directory.
4. Mount the new filesets on the temporary mount point.
5. Use the `vdump` command to copy the filesets from the old device. Use the `vrestore` command to restore them to the newly mounted filesets.
6. Unmount the old and new filesets.
7. Rename the new domain to the old name. Since you have not changed the domain and fileset names, it is not necessary to edit the `/etc/fstab` file.
8. Mount the new filesets using the mount points of the old filesets. The directory tree will then be unchanged. Delete the temporary mount point directory.

If you are running Version 5.0 or later, the new domain is created with the new DVN of 4 (see Section 1.4.3). However, if you must retain the DVN of 3 in order to use earlier versions of the operating system, see `mkfdmn(8)`. The `vdump` and `vrestore` utilities are not affected by the change of DVN.

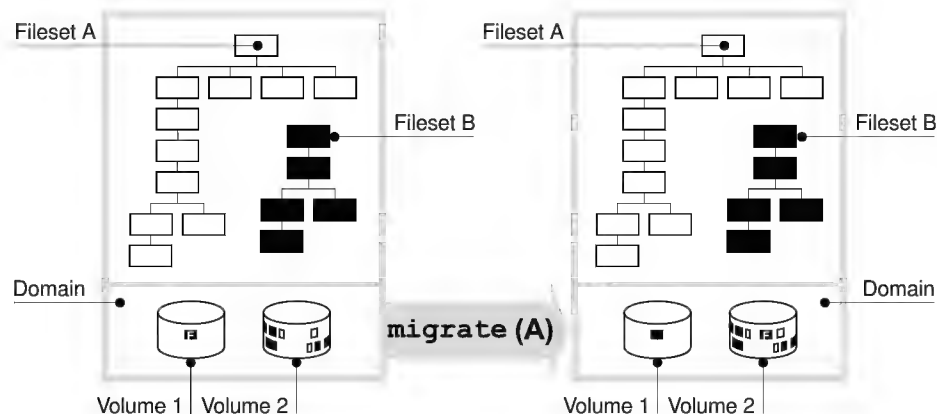
The following example assumes you have only one volume and moves the domain `accounts` with the fileset `technical` to volume `dsk3c` using the same fileset names. The domain `new_accounts` is the temporary domain. Assume the fileset `accounts#technical` is mounted on `/technical`. Assume that the `/etc/fstab` file has an entry to mount `accounts#technical` on `/technical`.



```
# mkfdmn /dev/disk/dsk3c new_accounts
# mkfset new_accounts technical
# mkdir /tmp_mnt
# mount new_accounts#technical /tmp_mnt
# vdump -dx - /technical|vrestore -xf - -D /tmp_mnt
# umount /technical
# umount /tmp_mnt
# rmfdmn accounts
# mv /etc/fdmns/new_accounts/ /etc/fdmns/accounts/
# mount /technical
# rmdir /tmp_mnt
```

If you have the optional AdvFS Utilities, you can use the `migrate` utility to move heavily accessed or large files to a different volume in the domain. The `balance` and `defragment` utilities also migrate files but are not under user control. With the `migrate` command, you can specify the volume where a file is to be moved or allow the system to pick the best space in the domain. You can migrate either an entire file or specific pages to a different volume. Figure 4-3 illustrates the `migrate` process.

**Figure 4-3: Migrating Files**



To move an entire file to a specific volume, use the `migrate` command with the `-d` option:

```
migrate -s -d destination_vol_index file_name
```

A file that is migrated will be defragmented in the process if possible. This means that you can use the `migrate` command to defragment selected files.

The following restrictions apply to the `migrate` utility:

- You must have root user privilege.
- You can only perform one migrate operation at a time on the same file.

- When you migrate a striped file, you can only migrate from one volume at a time.
- The `migrate` utility does not evaluate your migration decisions. For example, you can move more than one striped file segment to the same disk, which defeats the purpose of striping the file.

#### 4.3.3.1 Choosing to Migrate

Choose the `migrate` utility over the `balance` utility when you want to control the files that are moved. The `balance` utility moves files only to optimize distribution. For example, it might move many small files when moving a single larger one would be a better solution for your system.

Choose the `migrate` utility over the `defragment` utility when you want to defragment an individual file. If you have a large enough contiguous area on disk, you can migrate the file to that area to defragment it.

You can use the `showfile -x` command to look at the extent map and the performance percentage of a file. A low performance percentage (less than 80%) indicates that the file is fragmented on the disk. The extent map shows whether the entire file or a portion of the file is fragmented.

#### 4.3.3.2 Migrate Example

The following example displays the extent map of a file called `src` and migrates the file. The file, which resides in a two-volume domain, shows a change from 11 file extents to one and a performance efficiency improvement from 18% to 100%:

```
# showfile -x src
  Id Vol PgSz Pages XtntType Segs SegSz I/O Perf File
8.8002 1 16 11 simple ** ** async 18% src
  extentMap: 1
    pageOff pageCnt vol volBlock blockCnt
        0      1    1    187296      16
        1      1    1    187328      16
        2      1    1    187264      16
        3      1    1    187184      16
        4      1    1    187216      16
        5      1    1    187312      16
        6      1    1    187280      16
        7      1    1    187248      16
        8      1    1    187344      16
        9      1    1    187200      16
       10      1    1    187232      16
  extentCnt: 11

# migrate -d 2 src
# showfile -x src
```

```

      Id Vol PgSz Pages XtntType Segs SegSz I/O  Perf  File
8.8002  1  16   11   simple  **   ** async 100%  src
  extentMap: 1
    pageOff    pageCnt      vol    volBlock    blockCnt
          0         11        2      45536         176
  extentCnt: 1

```

The file `src` now resides on volume 2, consists of one file extent, and has a 100% performance efficiency. Note that in the output above, the first data line of the display lists the metadata. The metadata does not migrate to the new volume. It remains in the original location. The `extentMap` portion of the display lists the migrated files.

You can tailor the `migrate` utility to the needs of your system. You can let the system pick a new location in the domain. You can migrate specified pages of a file or you can move the pages of a striped file to different volumes within a domain. See `migrate(8)` for a detailed examples.

#### 4.3.4 Striping Files

Striping distributes files across a number of volumes. This increases the sequential read/write performance because I/O requests to the different disk drives can be overlapped. Virtual storage solutions, such as LSM, RAID, and storage area networks (SAN), stripe whole systems and are usually configured at system setup. AdvFS striping is applied to single files and is executed any time.

Use AdvFS striping only on directly attached storage that does not include LSM, RAID, or a SAN. Combining AdvFS striping with system striping may conflict with optimal placement and cause system degradation.

The AdvFS `stripe` utility distributes file segments across specific disks (or volumes) of a domain. You must have the Advanced Utilities to run this command. The stripe width is fixed at 64 kilobytes, but you can specify the number of volumes over which to stripe the file.

To stripe a file, create a new, empty file. Stripe it, specifying the number of volumes over which it should be striped. If desired, copy the content of the old file to the new.

As the file is appended, AdvFS determines the number of pages per stripe segment; the segments alternate among the disks in a sequential pattern. For example, the file system allocates the first segment of a two-disk striped file on the first disk and the next segment on the second disk. This completes one sequence, or stripe. The next stripe starts on the first disk, and so on. Because AdvFS spreads the I/O of the striped file across the specified disks, the sequential read/write performance of the file increases.

To stripe a file, enter the `stripe` command:

```
stripe -n volume_count filename
```

You cannot use the `stripe` utility to modify the number of disks that an already striped file crosses or to restripe a file that is already striped. To change the configuration of a striped file, you must create a new file, stripe it, then copy the original file data to it.

You cannot stripe the `/etc/fstab` file.

#### 4.3.4.1 Choosing to Stripe an AdvFS File

Before you use the `stripe` utility, run the `iostat` utility either from the SysMan “View Input/Output (I/O) Statistics” or from the command line (see `iostat(1)`) to determine if disk I/O is causing the bottleneck. The blocks per second and transactions per second should be cross checked with the drive’s sustained transfer rate. If the disk access is slow, then striping is one of the ways to improve performance (see Section 5.3). Maximum stripe performance will be achievable if each stripe disk is on its own disk controller.

It is not advisable to use AdvFS striping when system-wide striping is in effect. This may degrade performance.

#### 4.3.4.2 AdvFS Stripe Example

To stripe a file,

1. Create an empty file and stripe it across the number of volumes desired.
2. Copy the data from the original file to the striped file.
3. Delete the original file and rename the striped file, if desired.

The following example creates an empty file, stripes it, copies data into the striped file, then shows the extents of the striped file:

```
# touch file_1
# ls -l file_1
-rw-r--r-- 1 root system 0 Oct 07 11:06 file_1
# stripe -n 3 file_1

# cp orig_file_1 file_1
#showfile -x file_1

      Id Vol PgSz Pages XtntType Segs SegSz I/O   Perf File
7.8001  1  16   71  stripe    3    8 async 100% file_1
  extentMap: 1
    pageOff  pageCnt  volIndex  volBlock  blockCnt
          0         8          2    42400      384
          24         8
          48         8
    extentCnt: 1
  extentMap: 2
    pageOff  pageCnt  volIndex  volBlock  blockCnt
          8         8          3    10896      384
          32         8
          56         8
    extentCnt: 1

  extentMap: 3
    pageOff  pageCnt  volIndex  volBlock  blockCnt
         16         8          1   186784      368
         40         8
         64         7
    extentCnt: 1
```

#### 4.3.4.3 Removing AdvFS Striping

You can alter the pattern of striping in your domain:

- Remove striping from a file

If you have a striped file that you no longer want to be striped, copy it to a file that is not striped. Delete the original.

- Removing a striped volume

If you remove a volume that contains an AdvFS stripe segment, the `rmvol` utility moves the segment to another volume that does not already contain a stripe segment of the same file. If all remaining volumes contain stripe segments, the system requests confirmation before the segment is moved to a volume that already contains a stripe segment of the file. To retain the full benefit of striping when this occurs, stripe a new file across existing volumes and copy the file with the doubled-up segments to it.

### 4.3.5 Data Cache Tuning

Caching improves performance when data is frequently reused. AdvFS uses a dynamic memory cache called the Unified Buffer Cache (UBC) for managing file metadata and user data.

Dynamic caching gives AdvFS the ability to cache data up to available memory. The UBC shrinks the cache size as other system demands require memory.

Cache size limits are set and adjusted by tunable parameters (see System Configuration and Tuning). There are also parameters that limit the number of dirty pages cached.

### 4.3.6 Changing Attributes to Improve System Performance

A number of attributes can be changed to improve system performance. System Configuration and Tuning details the significance of each and the trade-offs engendered when they are changed. See `sysconfig(8)` for more information. You can modify attributes to:

- Increase the dirty-data caching threshold.  
Dirty or modified data is data that has been written by an application and cached but has not yet been written to disk. You can modify the amount of dirty data that AdvFS will cache for each volume in a domain with the `chvol -t` command or for all new volumes of a file system with the `AdvfsReadyQLim` attribute (see `chvol(8)`).
- Promote continuous I/O with `smooth sync`.  
The `smooth sync` queue improves AdvFS asynchronous I/O performance; that is, it increases file system efficiency in writing modified pages to disk. The `smooth sync` functionality is controlled by the `vfs` attribute `smoothsync_age`. By default `smooth sync` is enabled on the system.
- Change the I/O transfer size.  
AdvFS reads and writes data by 8-kilobyte pages. The maximum transfer size depends on the underlying storage configuration but is typically 128 or 256 blocks. LSM may assign a larger maximum transfer size. The maximum transfer size is adjustable using the `chvol` command (see `chvol(8)`).
- Flush modified mmaped pages.  
The `AdvfsSyncMmapPages` attribute controls whether modified mmaped pages are flushed to disk during a `sync` system call.
- Increase the memory available for access structures.

AdvFS allocates access structures until the percentage of pageable memory used for the access structures is `AdvfsAccessMaxPercent`. Increasing the value of the `AdvfsAccessMaxPercent` attribute may improve AdvFS performance on systems that open and reuse many files, but this will decrease the memory available for the virtual memory subsystem and the Unified Buffer Cache (UBC). Decreasing the value of the attribute frees pageable memory but may degrade AdvFS performance on systems that open and reuse many files.

### 4.3.7 Controlling Domain Panic Information

The `AdvfsDomainPanicLevel` attribute allows you to choose whether to have crash dumps created when a domain panic occurs. Values of the attribute are:

- 0 – Create crash dumps for no domains.
- 1 – Create crash dumps only for domains with mounted filesets (default).
- 2 – Create crash dumps for all domains.
- 3 – Promote the domain panic to a system panic. The system will crash.

See `sysconfig(8)` for information on changing attributes. See Section 5.4.8 for information about recovering from a domain panic.

## 4.4 Using a Trashcan

If you have the optional AdvFS Utilities, end users can configure their systems to retain a copy of files they have deleted. Trashcan directories can be attached to one or more directories within the same fileset. Once attached, any file deleted from an attached directory is automatically moved to the trashcan directory. The last version of a file deleted from a directory with a trashcan attached can be returned to the original directory with the `mv` command.

Trashcan directories are a trade off, however. The convenience of recovering files without accessing backup comes at the cost of the additional writes to disk that are required when files are deleted.

Root user privilege is not required to use this command. However, the following restrictions apply:

- You can restore only the most recently deleted version of a file.
- You can attach more than one directory to the same trashcan directory; however, if you delete files with identical file names from the attached directories, only the most recently deleted file remains in the trashcan directory.

- Only files you delete directly are removed to the trashcan. If you delete a complete fileset using the `rmfset` command, the files in it are not saved.
- Deleted files in an attached trashcan count against your quota.
- When you delete files in the trashcan directory, they are unrecoverable.

Table 4-1 lists and defines the commands for setting up and managing a trashcan:

**Table 4-1: Trashcan Commands**

Command	Description
<code>mktrashcan</code>	Creates the trashcan.
<code>shtrashcan</code>	Shows the contents of the trashcan.
<code>rmtrashcan</code>	Removes the trashcan directory.

For example, to attach the trashcan directory `keeper` to the directory `booklist`:

```
# mkdir keeper
# mktrashcan keeper /booklist
  'keeper' attached to '/booklist'
```

To remove a file, and look for it in the trashcan directory:

```
# rm old_titles
# shtrashcan /booklist
  '//keeper' attached to '/booklist'
# cd keeper
# ls
  old_titles
```

To remove the connection between the trashcan and the directory:

```
# rmtrashcan /booklist
  '/booklist' detached
```



---

## Troubleshooting

This chapter examines problems that, while universal for file systems, may have unique solutions for AdvFS. See System Configuration and Tuning for related information about diagnosing performance problems.

### 5.1 Disk File Structure Incompatibility

If you install your Version 5 operating system as an update to your Version 4 system (not a full installation), your `/root`, `/usr`, and `/var` files will retain a DVN of 3 (see Section 1.4.3.1).

By default, domains created on Version 5.0 and later have a new format that is incompatible with earlier versions (see Section 1.4.3). The newer operating system recognizes the older disk structure, but the older does not recognize the newer. To access a fileset with the new format (a DVN of 4) from an older operating system, NFS mount the fileset from a Version 5 system or upgrade your operating system to Version 5. There is the potential for problems when files created on one operating system are moved to another.

If you try to mount a fileset belonging to a domain with a DVN of 4 when you are running a version of the operating system earlier than Version 5.0, you will get an error message.

There is no tool that upgrades all domains with a DVN of 3 to domains with DVN of 4. You must upgrade each domain (see Section 1.4.3.2).

#### 5.1.1 Utility Incompatibility

Because of the new on-disk file formats, some AdvFS-specific utilities from earlier releases have the potential to corrupt domains created using the new on-disk formats. All statically-linked AdvFS-specific utilities from earlier operating system versions will not run on Version 5.0 and later. These utilities are usually from operating system versions prior to Version 4.0. In addition, the following dynamically-linked utilities from earlier releases of Tru64 UNIX do not run on Version 5.0 and later:

- `advfsstat`
- `balance`
- `chvol`

- `defragment`
- `rmvol`
- `showfdmn`
- `verify`

### 5.1.2 Avoiding Metadata Incompatibility

If a system crashes or goes down unexpectedly, for example due to loss of power, after reboot AdvFS will perform recovery when the filesets that were mounted at the time of the crash are remounted. This recovery keeps the AdvFS metadata consistent and makes use of the AdvFS transaction log file.

Different versions of the operating system use different AdvFS log record types. Therefore, it is important that AdvFS recovery be done on the same version of the operating system as was running at the time of the crash. For example, if your system was running Version 5.1 when it crashed, do not reboot using Version 3.2G because the log records may be formatted differently from those saved by the Version 5.1 system.

To reboot without error using a different version of the operating system, cleanly unmount all filesets before rebooting. Note that if the system failed due to a system panic or an AdvFS domain panic, it is best to reboot using the original version of the operating system and then run the `verify` command to ensure that the domain is not corrupted. If it is not, it is then safe to reboot using a different version of the operating system. If running the `verify` command indicates that the domain has been corrupted, see Section 5.4.6.

## 5.2 Memory Mapping, Direct I/O and Data Logging Incompatibility

Memory mapping, atomic-write data logging and direct I/O are mutually exclusive. If a file is open in one of these modes, attempting to open the same file in one of the conflicting modes will fail. For more information see Section 4.1.4 and Section 4.1.5 and the `mmap(2)` reference page.

## 5.3 Handling Poor Performance

The performance of a disk depends upon the I/O demands upon it. If your domain is structured so that heavy access is focused on one volume, it is likely that system performance will degrade. Once you have determined the load balance, there are a number of ways to equalize the activity and increase throughput. See System Configuration and Tuning, command reference pages, and Chapter 4 for more complete information.

To discover the causes of poor performance, first check system activity (see Section 4.2). There are a number of ways to improve performance:

- Upgrade domains

DVN4 domains are indexed when a directory grows beyond a page, that is, about 200 files (see Section 1.4.3.2). Directories with more than 5000 files show the most benefit.

- Eliminate disk access incompatibility

If you have initiated direct I/O (which turns off caching) to read and write data to a file, any application that accesses the same file will also have direct I/O. This may prove inefficient (see Section 4.1.5 ).

- Defragment domains

As files grow, contiguous space on disk often is not available to accommodate new data, so files become fragmented. File fragmentation can reduce system performance because more I/O is required to read or write a file. Use the AdvFS GUI (see Chapter 6) or run the `defragment` utility from the command line (see Section 4.3.1).

If you have AdvFS Utilities, you can also:

- Balance a multivolume domain

System performance improves if you distribute files evenly over all your volumes. Files that are distributed unevenly can degrade system performance. Use the `balance` command to redistribute the files (see Section 4.3.2).

When a volume is added to a domain with the `addvol` command, all the files of the domain remain on the previously existing volume and the new one is empty. To even the file distribution, use the AdvFS GUI (see Chapter 6) or run the `balance` utility from the command line.

- Stripe individual files

AdvFS allows you to stripe individual files across multiple volumes (see Section 4.3.4). Use AdvFS striping only on directly attached storage that is not otherwise striped. Combining AdvFS striping with other striping may degrade performance.

- Migrate individual files

You can use the `migrate` utility to move a heavily accessed file or selected pages of a file to another volume in the domain. You can move the file to a specific volume or you can let the system choose (see Section 4.3.3).

- Change AdvFS resources

You can change your file system size in the following ways:

- Increase the size of a domain by adding a volume with the `addvol` command (see Section 1.4.6). For optimum performance, each volume you add should consist of the entire disk (typically, partition `c`). Do not add a volume containing any data you want to keep. When you run the `addvol` command, data on the added disk is destroyed.
- Shrink a domain by removing a volume with the `rmvol` command (see Section 1.4.7). Striped file segments will be moved to a volume that does not contain a stripe. If this is not possible, the system requests confirmation before doubling up on stripes (see Section 4.3.4).

You can interrupt the `rmvol` process with `Ctrl/C` without damaging your domain. Files already removed from the volume will remain in their new location. Files that had not been moved at the time of the interrupt will remain in their original location.

If the volume that has had the files removed does not allow new file allocations after an aborted `rmvol` operation, use the `chvol` command with the `-A` option to reactivate the volume.

- Change the size of a domain by changing volumes. Add a new one, move your files to it, then remove the old (see Section 4.3.3).

## 5.4 Handling Disk Problems

Back up your data regularly and frequently and watch for signs of impending disk failure. Removing files from a problem disk before it fails can prevent a lot of trouble. See the Event Management information in System Administration for more information.

### 5.4.1 Checking Free Space and Disk Usage

You can look at the way space is allocated on a disk by file, fileset, or domain. The AdvFS GUI (see Chapter 6) displays a hierarchical view of disk objects and the space they use. Table 5-1 shows command-line commands that examine disk space usage.

**Table 5-1: Disk Space Usage Information Commands**

Command	Description
<code>du</code>	Displays information about block allocation for files; use the <code>-a</code> option to display information for individual files.
<code>df</code>	Displays disk space usage by fileset; available space for a fileset is limited by the fileset quota if it is set.

**Table 5–1: Disk Space Usage Information Commands (cont.)**

<code>showfdmn</code>	Displays the attributes and block usage for each volume in an active domain; for multivolume domains, additional volume information is displayed.
<code>showfile</code>	Displays block usage and volume information for a file or for the contents of a directory.
<code>showfsets</code>	Displays information about the filesets in a domain; use to display fileset quota limits.
<code>vdf</code>	Displays used and available disk space for a fileset or a domain.

---

See the reference pages for the commands for more complete information.

Under certain conditions, the disk usage information for AdvFS may become corrupt. To correct this, change the entry in the `/etc/fstab` file to enable the `quotacheck` command to run. The `quotacheck` command only checks filesets that have the `userquota` and `groupquota` options specified. For example, for the fileset `usr_domain#usr`:

```
usr_domain#usr /usr advfs rw,userquota,groupquota 0 2
```

Then run the `quotacheck` command for the fileset:

```
# quotacheck usr_domain#usr
```

This should correct the disk usage information.

## 5.4.2 Reusing AdvFS Volumes

All volumes (disks, disk partitions, LSM volumes, etc.) are labeled either `unused` or with the file system for which they were last used. You can only add a volume labeled `unused` to your domain (see Section 1.3).

If the volume you wish to add is part of an existing domain (the `/etc/fdmns` directory entry exists), the easiest way to return the volume label to `unused` status is to remove the volume with the `rmvol` command or to remove the domain with the `rmfdmn` command (which labels all volumes that were in the domain `unused`).

For example, if your volume is `/dev/disk/dsk5c`, your original domain is `old_domain`, and the domain you want to add the volume to is `new_domain`, mount all the filesets in `old_domain` then enter:

```
# rmvol /dev/disk/dsk5c old_domain
# addvol /dev/disk/dsk5c new_domain
```

If the volume you want to add is not part of an existing domain but is giving you a warning message because it is labeled, reset the disk label. If you

answer **yes** to the prompt on the `addvol` or `mkfdmn` command, the disk label will be reset. You will lose all information that was on the volume that you are adding.

### 5.4.3 Dumping to Block 0

To dump to a partition that starts at block 0 of a disk, you must first clear the disk label. If you do not, the `vdump` command may appear to contain valid savesets, but when the `vrestore` command attempts to interpret the disk label as part of the saveset, it will return an error (see Section 3.1.5).

### 5.4.4 Disk Space Usage Limits

If your system has been running without any limits on resource usage, you can add quotas to limit the amount of disk space your users can access. AdvFS quotas provide a layer of control beyond that available with UFS.

User and group quotas limit the amount of space a user or group can allocate for a fileset. Fileset quotas restrain a fileset from grabbing all of the available space in a domain.

You can set two types of quota limits: hard limits that cannot be exceeded and soft limits that can be exceeded for a period of time called the grace period. You can turn quota enforcement on and off. See Chapter 2 for complete information.

If you are working in an editor and realize that the information you need to save will put you over your quota limit, do not abort the editor or write the file because data may be lost. Instead, remove files to make room for the edited file prior to writing it. You can also write the file to another fileset, such as `tmp`, remove files from the fileset whose quota you exceeded, and then move the file back to that fileset.

AdvFS will impose quota limits in the rare case that you are 8 kilobytes below the user, group, or fileset quota and are attempting to use some or all of the space you have left. This is because AdvFS allocates storage in units of 8 kilobytes. If adding 8 kilobytes to a file would exceed the quota limit, then that file cannot be extended.

### 5.4.5 Verifying File System Consistency

To ensure that metadata is consistent, run the `verify` command to verify the file system structure. This utility checks disk structures such as the bitfile metadata table (BMT), the storage bitmaps, the tag directory, and the `frag` file for each fileset. It verifies that the directory structure is correct, that all directory entries reference a valid file, and that all files have a directory entry. You must be the root user to run this command.

It is a good idea to run the `verify` command:

- When problems are evident (corruptions, domain panic, lost data, I/O errors).
- Before an update installation.
- If your files have not been accessed in three to six months or longer and you plan to run utilities such as `balance`, `defragment`, `migrate`, `quotacheck`, `repquota`, `rmfset`, `rmvol`, or `vdump` that access every file in a domain.

Use the SysMan “Repair an AdvFS Domain” or, from the command line, enter:

```
verify domain_name
```

The `verify` command mounts filesets in special directories as it proceeds. If the command is unable to mount a fileset due to the failure of a domain, as a last resort run the command with the `-F` option. This option mounts the fileset using the `-d` option of the `mount` command, which means that AdvFS initializes the transaction log for the domain without recovery. As no domain recovery will occur for previously incomplete operations, this could cause data corruption.

Under some circumstances the `verify` command may fail to unmount the filesets. If this occurs, you must unmount the affected filesets manually.

On machines with many millions of files, sufficient swap must be allocated for the `verify` utility to run to completion. If the amount of memory required by `verify` exceeds the kernel variable `proc/max_per_proc_data_size` process variable, the utility will not complete. To overcome this problem, allocate up to 10% of the domain size in swap for running the `verify` command.

The following example verifies the `domainx` domain, which contains the filesets `setx` and `sety`:

```
# verify domainx
+++Domain verification+++
Domain Id 2f03b70a.000f1db0
Checking disks ...
Checking storage allocated on disk /dev/disk/dsk10g
Checking storage allocated on disk /dev/disk/dsk10a
Checking mcell list ...
Checking mcell position field ...
Checking tag directories ...

+++ Fileset verification +++
+++ Fileset setx +++
Checking frag file headers ...
```

```

Checking frag file type lists ...
Scanning directories and files ...
    1100
Scanning tags ...
    1100
Searching for lost files ...
    1100

+++ Fileset sety +++
Checking frag file headers ...
Checking frag file type lists ...
Scanning directories and files ...
    5100
Scanning tags ...
    5100
Searching for lost files ...
    5100

```

In this example, the `verify` command finds no problems with the domain.

## 5.4.6 Salvaging File Data from a Damaged Domain

How you recover file data from a damaged domain depends on the severity of the damage. Pick the simplest recovery path for the information you have.

1. Run the `verify` utility to try to repair the domain (see Section 5.4.5 and `verify(8)`). The `verify` utility can only fix a limited set of problems.
2. Recreate the domain from your most recent backup.
3. If your backup is not recent enough, use your most recent backup with the `salvage` utility to obtain more current copies of files.

The amount of data you are able to recover will depend upon the damage to your domain. You must be root user to run the `salvage` utility. See `salvage(8)` for more information.

Use the SysMan “Recover Files from an AdvFS Domain” or, from the command line, enter:

```
salvage domain_name
```

Running the `salvage` utility does not guarantee that you will recover all of your domain. You may be missing files, directories, file names, or parts of files. The utility generates a log file that contains the status of files that were recovered. Use the `-l` option to list in the log file the status of all files that are encountered.

The `salvage` utility places recovered files in directories named after the filesets. There is a `lost+found` directory for each fileset that contains files for which no parent directory can be found. You can specify the path name of



the directory that is to contain the fileset directories. If you do not specify a directory, the utility writes recovered filesets under the current working directory. You cannot mount the directories in which the files are recovered. You must move the recovered files to new filesets.

The best way to recover your domain is to use your daily backup tapes. If files have changed since the last backup, you can use the tapes along with the `salvage` utility as follows:

1. Create a new domain and filesets to hold the recovered information. Mount the filesets.
2. Restore from your backup tape(s) to the new domain.
3. Run the `salvage` utility with the `-d` option set to recover files that have changed since the backup. If you have no backups, you can run the `salvage` utility without the `-d` option to recover all the files in the domain.

The fastest salvage process is to recover file information to another location on disk. The following example recovers data to disk:

```
# /sbin/advfs/salvage -d 199812071330 corrupt3_domain
salvage: Domain to be recovered 'corrupt3_domain'
salvage: Volume(s) to be used '/dev/disk/dsk12a'
        '/dev/disk/dsk12g' '/dev/disk/dsk12h'
salvage: Files will be restored to '.'
salvage: Logfile will be placed in './salvage.log'
salvage: Starting search of all filesets:
        09-Mar-2000 11:53:40
salvage: Starting search of all volumes:
        09-Mar-2000 11:55:41
salvage: Loading file names for all filesets:
        09-Mar-2000 11:56:42
salvage: Starting recovery of all filesets:
        09-Mar-2000 11:57:02
```

If not enough room is available on disk for the recovered information, you can recover data to tape and then write it back on to your original disk location. However, since this process destroys the original damaged data on disk, once you have created a new domain, there is no way to rerun the `salvage` command if problems arise.

1. Run the `salvage` command with the `-d` option set and use the `-F` and `-f` options to specify tar format and tape drive. If you have no backups, you can run the `salvage` utility without the `-d` option to recover all the files in the domain.
2. Remove the corrupt domain.

3. Create a new domain and filesets to hold the recovered information. Mount the filesets.
4. Restore from your backup tape(s) to the new domain.
5. Extract the `tar` archive from the tape that the `salvage` utility created (see `tar(1)`) to the new filesets.

---

### Caution

---

Writing over the corrupt data on the disk is an irreversible process. If there is an error, you can no longer recover any more data from the corrupt domain. Therefore, look at the `salvage` log file or the files on the `tar` tape to make sure you have gotten all the files you need. If you have not recovered a significant number of files, you can use the `salvage` command with the `-s` option described below.

---

The following example recovers data to tape and restores the data to a newly created domain:

```
# /sbin/advfs/salvage -F tar -d 9810280930 corrupt_domain
salvage: Domain to be recovered 'corrupt_domain'
salvage: Volume(s) to be used '/dev/disk/dsk8c'
        '/dev/disk/dsk5c'
salvage: Files will archived to '/dev/tape/tape0_d1'
        in TAR format
salvage: Logfile will be placed in './salvage.log'
salvage: Starting search of all filesets:
        09-Mar-2000 10:28:13
salvage: Starting search of all volumes:
        09-Mar-2000 10:31:41
# rmfdmn corrupt_domain
# mkfdmn /dev/disk/dsk5c good_domain
# addvol /dev/disk/dsk8c good_domain
# mkfset good_domain fset1
# mkfset good_domain fset2
# mount good_domain#fset1 /fset1
# mount good_domain#fset2 /fset2
```

Then restore filesets from tape(s) created by the `salvage` command.

```
# cd /fset1
# tar -xpf /dev/tape/tape0_d1 fset1
# cd /fset2
# tar -xpf /dev/tape/tape0_d1 fset2
```

If you have run the `salvage` utility and have been unable to recover a large number of files, run `salvage` with the `-s` option set. This process is very slow because the utility reads every disk block at least once.

---

### Caution

---

The `salvage` utility with the `-s` option set opens and reads block devices directly. This could present a security problem. It may be possible to recover data from older, deleted AdvFS domains while attempting to recover data from current AdvFS domains.

---

Note that if you have chosen recovery to tape and have already created a new domain on the disks containing the corrupted domain, you cannot use the `-s` option because your original information has been lost.

---

### Note

---

If you have accidentally used the `mkfdomn` command on a good domain, running the `salvage` utility with the `-s` option set is the only way to recover files.

---

For example:

```
# salvage -s corrupt3_domain
salvage: Domain to be recovered 'corrupt3_domain'
salvage: Volume(s) to be used '/dev/disk/dsk2a'
        '/dev/disk/dsk2g' '/dev/disk/dsk2h'
salvage: Files will be restored to '.'
salvage: Logfile will be placed in './salvage.log'
salvage: Starting sequential search of all volumes:
        08-May-2000 14:45:39
salvage: Loading file names for all filesets:
        08-May-2000 15:00:38
salvage: Starting recovery of all filesets:
        08-May-2000 15:00:40
```

## 5.4.7 “Can’t Clear a Bit Twice” Error Message

If you receive a “Cannot clear a bit twice” error message, your domain is damaged. To repair it:

1. Set the `AdvfsFixUpSBM` kernel variable to allow access to the damaged domain. This flag is off by default
2. Mount and back up the filesets in the damaged domain.
3. Turn `AdvfsFixUpSBM` off.
4. Unmount the filesets in the domain Run the `verify` utility with the `-f` option. If there are errors, continue through steps 5 and 6.
5. Recreate the domain and filesets.
6. Restore from the backup.

To turn AdvfsFixUpSBM on:

```
# dbx -k /vmunix /dev/mem
dbx> assign AdvfsFixUpSBM = 1
dbx> quit
```

To turn AdvfsFixUpSBM off:

```
# dbx -k /vmunix /dev/mem
dbx> assign AdvfsFixUpSBM = 0
dbx> quit
```

---

**Note**

---

The AdvfsFixUpSBM variable is global.  
Turn it off so that the error message is again available for all domains.

---

## 5.4.8 Recovering from a Domain Panic

When a metadata write error occurs, or if corruption is detected in a single AdvFS domain, the system initiates a domain panic (rather than a system panic) on the domain. This isolates the failed domain and allows a system to continue to serve all other domains. After a domain panic AdvFS no longer issues I/O requests to the disk controller for the affected domain. Although the domain cannot be accessed, the filesets in the domain can be unmounted.

When a domain panic occurs, an EVM event is logged (see EVM(5)) and the following message is printed to the system log and the console:

**AdvFS Domain Panic; Domain name Id domain\_Id**

For example:

```
AdvFS Domain Panic; Domain staffb_domain Id 2dad7c28.0000dfbb
An AdvFS domain panic has occurred due to either a
  metadata write error or an internal inconsistency.
This domain is being rendered inaccessible.
```

By default, a domain panic on an active domain will cause a live dump to be created and placed in the /var/adm/crash directory. Please file a problem report with your software support organization and include the dump file and a copy of the running kernel.

To recover from a domain panic, perform the following steps:

1. Run the `mount` command with the `-t` option and identify all mounted filesets in the affected domain.

2. Unmount all these filesets.
3. Examine the `/etc/fdmns` directory to obtain a list of the AdvFS volumes in the domain that panicked.
4. Run the `savemeta` command (see `savemeta(8)`) to collect information about the metadata files for each volume in the domain for Compaq support personnel. These saved files will be written in the directory specified and contain information that technical support needs.
5. If the problem is a hardware problem, fix it before continuing.
6. Run the `verify` utility on the domain (see Section 5.4.5).
  - If there are no errors, mount all the filesets you unmounted and resume normal operations.
  - If the `verify` command was able to run but showed errors, mount the filesets, do a backup, and recreate the domain. Note that the backup may be incomplete and that earlier backup resources may be needed.
7. If the failure prevents complete recovery, recreate the domain with the `mkfdmn` command and restore the domain's data from backup. If this does not provide enough information, you may need to run the `salvage` utility (see Section 5.4.6).

For example:

```
# mount -t advfs
staffb_dmn#staff3_fs on /usr/staff3 type advfs (rw)
staffb_dmn#staff4_fs on /usr/staff4 type advfs (rw)
# umount /usr/staff3
# umount /usr/staff4
# ls -l /etc/fdmns/staffb_dmn
lrwxr-xr-x 1 root system 10 Aug 25 16:46
dsk35c->/dev/disk/dsk3c
lrwxr-xr-x 1 root system 10 Aug 25 16:50
dsk36c->/dev/disk/dsk6c
lrwxr-xr-x 1 root system 10 Aug 25 17:00
dsk37c->/dev/disk/dsk1c
# /sbin/advfs/savemeta staffb_dmn /tmp/saved_dmn
# verify staffb_dmn
```

You do not need to reboot after a domain panic.

If you have recurring domain panics, it may be helpful to adjust the `AdvfsDomainPanicLevel` attribute (see Section 4.3.7) in order to facilitate debugging.

### 5.4.9 Recovering from Filesets Mounted Read-Only

When a fileset is mounted, AdvFS verifies that all volumes in a domain can be accessed. The size recorded in the domain's metadata for each volume must match the size of the volume. If the sizes match, the mount proceeds. If a volume is smaller than the recorded size, AdvFS attempts to read the last block marked in use for the fileset. If this block can be read, the mount will succeed, but the fileset will be marked as read-only. If the last in-use block for any volume in the domain cannot be read, the mount will fail. See `mount(8)` for more information.

If a fileset is mounted read-only, check the labels of the flagged volumes in the error message. There are two common errors:

- A disk is mislabeled on a RAID array.
- An LSM volume upon which an AdvFS domain resides has been shrunk from its original size (see Section 1.7).

If you have AdvFS Utilities and if the domain consists of multiple volumes and has enough free space to remove the offending volume, you do not need to remove your filesets. However, it is a good idea to back them up before proceeding:

1. Remove the volume from the domain using the `rmvol` command. (This will automatically migrate the data to the remaining volumes.)
2. Correct the disk label of the volume with the `disklabel` command.
3. Add the corrected volume back to the domain with the `addvol` command.
4. Run the `balance` command to distribute the data across the new volumes.

For example, if `/dev/disk/dsk2c` (on a device here called `<disk>`) within the `data5` domain is mislabeled, you can migrate your files on that volume (automatic with the `rmvol` command), then move them back when you have restored the volume:

```
# rmvol /dev/disk/dsk2c data5
# disklabel -z dsk2
# disklabel -rw dsk2 <disk>
# addvol /dev/disk/dsk2c data5
# balance data5
```

If you do not have AdvFS Utilities or if there is not enough free space in the domain to transfer the data from the offending volume:

1. Back up all filesets in the domain.
2. Remove the domain with the `rmfdmn` command.
3. Correct the disk label of the volume with the `disklabel` command.
4. Make the new domain.
5. If you have AdvFS Utilities and if the original domain was multivolume, add the corrected volume back to the domain with the `addvol` command.
6. Restore the filesets from the backup.

For example, if `/dev/disk/dsk1c` (on a device here called `<disk>`) containing the `data3` domain is mislabeled:

```
# vdump -0f -u /data3
# rmfdmn data3
# disklabel -z dsk1 <disk>
# disklabel -w dsk1 <disk>
# mkfdmn data3
```

If you are recreating a multivolume domain, include the necessary `addvol` commands to add the additional volumes. For example to add `/dev/disk/dsk5c` to the domain:

```
# addvol /dev/disk/dsk5c data3
# mkfset data3 data3fset
# mount data3#data3fset /data3
# vrestore -xf - /data3
```

## 5.5 Restoring an AdvFS File System

Use the `vrestore` command to restore your AdvFS files that have been backed up with the `vdump` command.

### 5.5.1 Restoring the `/etc/fdmns` Directory

AdvFS must have a current `/etc/fdmns` directory in order to mount filesets (see Section 1.4.2). A missing or damaged `/etc/fdmns` directory prevents access to a domain, but the data within the domain remains intact. You can restore the `/etc/fdmns` directory from backup or you can recreate it.

If you have a current backup copy of the directory, it is preferable to restore the `/etc/fdmns` directory from backup. Any standard backup facility (`vdump`, `tar`, or `cpio`) can back up the `/etc/fdmns` directory. To restore the directory, use the recovery procedure that is compatible with your backup process.

You can reconstruct the `/etc/fdmns` directory manually or with the `advscan` command. The procedure for reconstructing the `/etc/fdmns` directory is similar for both single-volume and multivolume domains. You can construct the directory for a missing domain, missing links, or the whole directory.

If you choose to reconstruct the directory manually, you must know the name of each domain and its associated volumes.

#### 5.5.1.1 Reconstructing the `/etc/fdmns` Directory Manually

If you accidentally lose all or part of your `/etc/fdmns` directory, and you know which domains and links are missing, you can reconstruct it manually.

The following example reconstructs the `/etc/fdmns` directory and two domains where the domains exist and their names are known. Each contains a single volume (or special device). Note that the order of creating the links in these examples does not matter. The domains are:

```
domain1 on /dev/disk/dsk1c
```

```
domain2 on /dev/disk/dsk2c
```

To reconstruct the two single-volume domains, enter:

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c dsk1c
# mkdir /etc/fdmns/domain2
# cd /etc/fdmns/domain2
# ln -s /dev/disk/dsk2c dsk2c
```

The following example reconstructs one multivolume domain. The `domain1` domain contains the following three volumes:

```
/dev/disk/dsk1c
```

```
/dev/disk/dsk2c
```

```
/dev/disk/dsk3c
```

To reconstruct the multivolume domain, enter the following:

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c dsk1c
# ln -s /dev/disk/dsk2c dsk2c
# ln -s /dev/disk/dsk3c dsk3c
```



### 5.5.1.2 Reconstructing the /etc/fdmns Directory Using advscan

You can use the `advscan` command to determine which partitions on a disk or Logical Storage Manager (LSM) disk group are part of an AdvFS domain. Then you can use the command to rebuild all or part of your `/etc/fdmns` directory. This command is useful:

- When disks have moved to a new system, device numbers have changed, or you have lost track of a domain location.
- For repair, if you delete the `/etc/fdmns` directory, delete a domain from the `/etc/fdmns` directory, or delete links from a domain's subdirectory in the `/etc/fdmns` directory.

The `advscan` command can:

- Determine if a partition is an AdvFS partition.
- List partitions in the order they are found on disk.
- Read the disk label to determine which partitions are in the domain and if any are overlapping.
- Scan all disks found in any `/etc/fdmns` domain.
- Recreate missing domain directories. The domain name is created from the device name.
- Fix the domain count and links for a domain.

For each domain there are three numbers that must match for the AdvFS file system to operate properly:

- The number of physical partitions found by the `advscan` command that have the same domain ID
- The domain volume count (the number stored in the AdvFS metadata that specifies how many partitions the domain has)
- The number of `/etc/fdmns` links to the partitions, because each partition must be represented by a link

See `advscan(8)` for more information.

Inconsistencies can occur in these numbers in a number of ways and for a number of reasons. In general, the `advscan` command treats the domain volume count as more reliable than the number of partitions or `/etc/fdmns` links. The following tables list anomalies, possible causes, and corrective actions that `advscan` can take. In the table, the letter N represents the value that is expected to be consistent for the number of partitions, domain volume count, and number of links.

Table 5-2 shows possible cause and corrective action if the expected value, N, for the number of partitions and for the domain value count do not equal the number of links in `/etc/fdmns/<dmn>`.

**Table 5-2: Fileset Anomalies and Corrections**

Number of Links in <code>/etc/fdmns/&lt;dmn&gt;</code>	Possible Cause	Corrective Action
<N	<code>addvol</code> terminated early or a link in <code>/etc/fdmns/&lt;dmn&gt;</code> was manually removed.	If the domain is activated before running <code>advscan</code> with the <code>-f</code> option and the cause of the mismatch was an interrupted <code>addvol</code> , the situation will be corrected automatically. Otherwise, <code>advscan</code> will add the partition to the <code>/etc/fdmns/&lt;dmn&gt;</code> directory.
>N	<code>rmvol</code> terminated early or a link in <code>/etc/fdmns/&lt;dmn&gt;</code> was manually added.	If the domain is activated and the cause of the mismatch was an interrupted <code>rmvol</code> , the situation will be corrected automatically. Otherwise, if the cause was a manually added link in <code>/etc/fdmns/&lt;dmn&gt;</code> , systematically try removing different links in the <code>/etc/fdmns/&lt;dmn&gt;</code> directory and try activating the domain. The number of links to remove is the number of links in the <code>/etc/fdmns/&lt;dmn&gt;</code> directory minus the domain volume count displayed by <code>advscan</code> .

Table 5-3 shows possible cause and corrective action if the expected value, N, for the number of partitions and for the number of links in `/etc/fdmns/<dmn>` do not equal the domain volume count:

**Table 5-3: Fileset Anomalies and Corrections**

Domain Volume Count	Possible Cause	Corrective Action
<N	Cause unknown	Cannot correct; run <code>salvage</code> to recover as much data as possible from the domain.
>N	<code>addvol</code> terminated early and partition being added is missing or has been reused.	Cannot correct; run <code>salvage</code> to recover as much data as possible from the remaining volumes in the domain.

Table 5-4 shows possible cause and corrective action if the expected value, N, for the domain volume count and for the number of links in `/etc/fdmns/<dmn>` do not equal the number of partitions:

**Table 5–4: Fileset Anomalies and Corrections**

Number of Partitions	Possible Cause	Corrective Action
<N	Partition missing.	Cannot correct; run <code>salvage</code> to recover as much data as possible from the remaining volumes in the domain.
>N	<code>addvol</code> terminated early.	None; domain will mount with N volumes; rerun <code>addvol</code>

To locate AdvFS partitions, enter the `advscan` command:

```
advscan [options] disks
```

In the following example there are no missing domains. The `advscan` command scans devices `dsk0` and `dsk5` for AdvFS partitions and finds nothing amiss. There are two partitions found (`dsk0c` and `dsk5c`), the domain volume count reports two, and there are two links entered in the `/etc/fdmns` directory.

```
# advscan dsk0 dsk5
Scanning disks  dsk0 dsk5
Found domains:
usr_domain
          Domain Id      2e09be37.0002eb40
          Created        Thu Feb 24 09:54:15 2000
          Domain volumes      2
          /etc/fdmns links    2
          Actual partitions found:
                                dsk0c
                                dsk5c
```

In the following example, directories that define the domains that include `dsk6` were removed from the `/etc/fdmns` directory. This means that the number of `/etc/fdmns` links, the number of partitions, and the domain volume counts are no longer equal.

The `advscan` command scans device `dsk6` and recreates the missing domains as follows:

1. A partition is found containing an AdvFS domain. The domain volume count reports one, but there is no domain directory in the `/etc/fdmns` directory that contains this partition.
2. Another partition is found containing a different AdvFS domain. The domain volume count is also one. There is no domain directory that contains this partition.
3. No other AdvFS partitions are found. The domain volume counts and the number of partitions found match for the two discovered domains.

4. The `advscan` command creates directories for the two domains in the `/etc/fdmns` directory.
5. The `advscan` command creates symbolic links for the devices in the `/etc/fdmns` domain directories.

The command and output are as follows:

```
# advscan -r dsk6
Scanning disks dsk6
Found domains:
*unknown*
          Domain Id      2f2421ba.0008c1c0
          Created        Thu Jan 20 13:38:02 2000

          Domain volumes      1
          /etc/fdmns links    0

          Actual partitions found:
                              dsk6a*

*unknown*
          Domain Id      2f535f8c.000b6860
          Created        Fri Feb 25 09:38:20 2000

          Domain volumes      1
          /etc/fdmns links    0

          Actual partitions found:
                              dsk6b*

Creating /etc/fdmns/domain_dsk6a/
linking dsk6a

Creating /etc/fdmns/domain_dsk6b/
linking dsk6b
```

## 5.5.2 Recovering from Volume Failure

Some problems show up in AdvFS because of hardware errors. For example, if a write to the file system fails due to a hardware fault, it might show up as metadata corruption. Hardware problems cannot be repaired by your file system. If you start seeing unexplained errors from a file system, do the following:

1. As root user, examine the `/var/adm/messages` file for AdvFS I/O error messages. For example:

```
Sep 28 15:39:16 systemname vmunix: AdvFS I/O error:
Sep 28 15:39:16 systemname vmunix: Domain#Fileset:test1#test1
Sep 28 15:39:16 systemname vmunix: Mounted on: /test1
Sep 28 15:39:17 systemname vmunix: Volume: /dev/rz11c
```

```

Sep 28 15:39:17 systemname vmunix: Tag: 0x00000006.8001
Sep 28 15:39:17 systemname vmunix: Page: 76926
Sep 28 15:39:17 systemname vmunix: Block: 5164080
Sep 28 15:39:17 systemname vmunix: Block count: 256
Sep 28 15:39:17 systemname vmunix: Type of operation: Read
Sep 28 15:39:17 systemname vmunix: Error: 5
Sep 28 15:39:17 systemname vmunix: To obtain the name of
Sep 28 15:39:17 systemname vmunix: the file on which the
Sep 28 15:39:17 systemname vmunix: error occurred, type the
Sep 28 15:39:17 systemname vmunix: command
Sep 28 15:39:17 systemname vmunix: /sbin/advfs/tag2name
Sep 28 15:39:17 systemname vmunix: /test1/.tags/6

```

This error message describes the domain, fileset, and volume on which the error occurred. It also describes how to find out what file was affected by the I/O error. If you do not find any AdvFS I/O error messages but are still seeing unexplained behavior on the file system, unmount the domain as soon as possible and run the `verify` utility to check the consistency of the domain's metadata.

2. Check for device driver error messages for the volume described in the AdvFS I/O error message. If you do not find any error messages, unmount the domain as soon as possible and run the `verify` utility to check the integrity of the domain's metadata. If you do find device driver I/O error messages that correspond to the AdvFS I/O error messages, then the file system is being affected by problems with the underlying hardware.
3. Try to remove the faulty volume using the `rmvol` utility (see Section 1.4.7). If this succeeds, the file system problems should not recur. If `rmvol` fails due to more I/O errors, it will be necessary to recreate the domain.
4. If you have a recent backup, recreate the domain and restore it from backup. If you have no backup or it is too old, use the `salvage` utility (see Section 5.4.6) to extract the contents of the corrupted domain.
5. Remove the faulty domain using the `rmfdmn` command.
6. Recreate the domain using the `mkfdmn` command. Remember that if you are recreating your domain under Version 5.0 and later, your domains will have a DVN of 4 by default (see Section 1.4.3). Add volumes as needed if you have the AdvFS Advanced Utilities package installed. Do not include the faulty volume in the new domain.
7. Restore the contents of the recreated domain using the information obtained in step 4.
8. Remount the filesets in the domain.

### 5.5.3 Recovering from Failure of the root Domain

A catastrophic failure of the disk containing your AdvFS root domain requires that you recreate your root domain in order to have a domain to boot from. Before you recreate your domain, it is a good idea to satisfy yourself that the failure is not due to hardware problems. Check the console, look for cable or power problems, etc.

If you have files in the root domain that were not backed up, run the `salvage` utility with the `-d` option to obtain more recent information from your domain. Make sure that regularly scheduled jobs are disabled. Then boot from your installation CD-ROM.

To recover from the failure of the root domain:

1. Run the `salvage` utility if necessary and save the files at another location.
2. Boot your system as stand-alone.
3. Transfer to single-user mode.
4. Examine the devices available.
5. Label the disk you have chosen.
6. Create the root domain and fileset. Note that if you have changed the root domain name or fileset name, use the new name.
7. Mount the newly created root domain and restore from backup.
8. If necessary, move any files recovered from the `salvage` process into the root domain.
9. If necessary, move your `/usr` file to this disk.

The following example assumes that you are booting from the CD-ROM device DKA500, which is the installation Stand Alone System (SAS). The tape drive is `/dev/tape/tape0`. The root is being restored to device `/dev/disk/dsk1`, which is a device here called `<disk>`. The example boots in single-user mode, creates a new root domain, and restores its contents from backup.

```
>>> b DKA500
3) UNIX Shell
# ls /dev/disk
# ls /dev/tape/tape0
# disklabel -rw -t advfs /dev/rdisk/dsk1a <disk>
# mkfdmn -r /dev/disk/dsk1a root_domain
# mkfset root_domain root
# mount root_domain#root /mnt
# cd /mnt
# vrestore -x -D .
```

```
# mkfdmn /dev/disk/dsk1a usr_domain
# mkfset usr_domain usr
# mount usr/_domain#usr /usr
# mount root_domain#root /mnt
# cd /usr
```

You can now boot your restored root domain.

### 5.5.4 Restoring a Multivolume usr Domain

To restore a multivolume `/usr` file system, the `usr_domain` domain must first be reconstructed with all of its volumes before you restore the files. However, creating a multivolume domain requires the `addvol` utility, and the `addvol` command will not run unless the License Management Facility (LMF) database, which resides in the `/usr/sbin` directory, is available. See `lmf(8)` for information.

On some systems the `/var` directory, where the LMF database resides, and the `/usr` directory are both located in the `usr` fileset. So the directory containing the license database must be recovered from the `usr` fileset before the `addvol` command can be accessed. On some systems the `/var` directory is in a separate fileset. If this is the case, the `addvol` command can be recovered first and then can be used to add the volumes.

The following example restores a multivolume domain where the `/var` directory and the `/usr` directory are both in the `usr` fileset in the `usr_domain` domain consisting of the `dsk1g`, `dsk2c`, and `dsk3c` volumes. The procedure assumes that the root file system has already been restored.

1. Mount the root fileset as read/write:

```
# mount -u /
```

2. Remove the links for the old `usr_domain` and create a new `usr_domain` using the initial volume:

```
# rm -rf /etc/fdmns/usr_domain
# mkfdmn /dev/disk/dsk1g usr_domain
```

3. Create and mount the /usr and /var filesets:

```
# mkfset usr_domain usr# mount -t advfs usr_domain#usr /usr
```

4. Create a soft link in /usr because that is where the lmf command looks for its database:

```
# ln -s /var /usr/var
```

5. Insert the /usr backup tape:

```
# cd /usr
# vrestore -vi
(/) add sbin/addvol
(/) add sbin/lmf
(/) add var/adm/lmf
(/) extract
(/) quit
```

6. Reset the license database:

```
# /usr/sbin/lmf reset
```

7. Add the extra volumes to usr\_domain:

```
# /usr/sbin/addvol /dev/disk/dsk2c usr_domain
# /usr/sbin/addvol /dev/disk/dsk3c usr_domain
```

8. Do a full restore of the /usr backup:

```
# cd /usr
# vrestore -xv
```

The following example restores a multivolume domain where the /usr and /var directories are in separate filesets in the same multivolume domain, usr\_domain, consisting of dsk1g, dsk2c, and dsk3c. This means that you must mount both the /var and the /usr backup tapes. The procedure assumes that the root file system has already been restored.

1. Mount the root fileset as read/write:

```
# mount -u /
```

2. Remove the links for the old usr\_domain and create a new usr\_domain using the initial volume:

```
# rm -rf /etc/fdmns/usr_domain
# mkfdmn /dev/disk/dsk1g usr_domain
```

3. Create and mount the /usr and /var filesets:

```
# mkfset usr_domain usr
# mkfset usr_domain var
# mount -t advfs usr_domain#usr /usr
# mount -t advfs usr_domain#var /var
```



4. Insert the `/var` backup tape and restore from it:

```
# cd /var
# vrestore -vi
(/) add adm/lmf
(/) extract
(/) quit
```

5. Insert the `/usr` backup tape:

```
# cd /usr
# vrestore -vi
(/) add sbin/addvol
(/) add sbin/lmf
(/) extract
(/) quit
```

6. Reset the license database:

```
# /usr/sbin/lmf reset
```

7. Add the extra volumes to `usr_domain`:

```
# /usr/sbin/addvol /dev/disk/dsk2c usr_domain
# /usr/sbin/addvol /dev/disk/dsk3c usr_domain
```

8. Do a full restore of `/usr` backup:

```
# cd /usr
# vrestore -xv
```

9. Insert the `/var` backup tape and do a full restore of `/var` backup:

```
# cd /var
# vrestore -xv
```

## 5.6 Recovering from a System Crash

As each domain is mounted after a crash, it automatically runs recovery code that checks through the transaction log to ensure that any file system operations that were occurring when the system crashed are either completed or backed out. This ensures that AdvFS metadata is in a consistent state after a crash.

### 5.6.1 Saving Copies of System Metadata

If you believe that a domain is corrupted or otherwise causing problems, run the `savemeta` command to save a copy of the domain's metadata for examination by Compaq support personnel. You must be root user to run this command (see `savemeta(8)`).

## 5.6.2 Physically Moving an AdvFS Disk

If a machine has failed, it is possible to move disks containing AdvFS domains to another computer running AdvFS. Connect the disk(s) to the new machine and modify the `/etc/fdmns` directory so the new system will recognize the transferred volume(s). You must be root user to complete this process.

You cannot move domains that have a DVN of 4 to systems running a Version 4 operating system. Doing so will generate an error message (see Section 5.1). You can move domains with a DVN of 3 to a machine running Version 5. The newer operating system will recognize the domains created earlier.

---

### Caution

---

Do not use either the `addvol` command or the `mkfdmn` command to add the volumes to the new machine. Doing so will delete all data on the disk you are moving. See Section 5.4.6 if you have already done so.

---

If you do not know what partitions your domains were on, you can add the disks on the new machine and run the `advscan` command, which may be able to recreate this information. You can also look at the disk label on the disk to see which partitions in the past have been made into AdvFS partitions. This will not tell you which partitions belong to which domains.

For example, if the motherboard of your machine fails, you need to move the disks to another system. You may need to reassign the disk SCSI IDs to avoid conflicts. (See your disk manufacturer instructions for more information.) For this example, assume the IDs are assigned to disks 6 and 8. Assume also that the system has a domain, `testing_domain`, on two disks, `dsk3` and `dsk4`. This domain contains two filesets: `sample1_fset` and `sample2_fset`. These filesets are mounted on `/data/sample1` and `/data/sample2`.

Assume you know that the domain that you are moving had partitions `dsk3c`, `dsk4a`, `dsk4b`, and `dsk4g`. The moving process would take the following steps:

1. Shut down the working machine to which you are moving the disks.
2. Connect the disks from the bad machine to the good one.
3. Reboot. You do not need to reboot to single-user mode; multiuser mode works because you can complete the following steps while the system is running.

4. Figure out the device nodes created for the new disks:

```
# /sbin/hwmmgr -show scsi -full
```

The output is a detailed list of information about all the disks on your machine. The DEVICE FILE column shows the name that the system uses to refer to each disk. Determine the listing for the disk you just added, for example, `disk6`. Use this name to set up symbolic links in step 5 below.

5. Modify your `/etc/fdmns` directory to include the information from the transferred domains:

```
# mkdir -p /etc/fdmns/testing_domain
# cd /etc/fdmns/testing_domain
# ln -s /dev/disk/dsk6c dsk6c
# ln -s /dev/disk/dsk8a dsk8a
# ln -s /dev/disk/dsk8b dsk8b
# ln -s /dev/disk/dsk8g dsk8g
# mkdir /data/sample1
# mkdir /data/sample2
```

6. Edit the `/etc/fstab` file to add the fileset mount-point information:

```
testing_domain#sample1_fset /data/sample1 advfs rw 1 0
testing_domain#sample2_fset /data/sample2 advfs rw 1 0
```

7. Mount the volumes:

```
# mount /data/sample1
# mount /data/sample2
```

Note that if you run the `mkfdmn` command or the `addvol` command on partition `dsk6c`, `dsk8a`, `dsk8b`, or `dsk8g`, or an overlapping partition, you will destroy the data on the disk. See Section 5.4.6 if you have accidentally done so.

### 5.6.3 Log File Inconsistency

If a system crashes, AdvFS will perform recovery at reboot. Filesets that were mounted at the time of the crash will be recovered when they are remounted. This recovery keeps the AdvFS metadata consistent and makes use of the AdvFS transaction log.

Since different versions of the operating system use different transaction log structures, it is important that you recover your filesets on the version of the operating system that was running at the time of the crash. If you do not, you risk corrupting the domain metadata and/or panicking the domain.

If the system crash has occurred because you have set the `AdvfsDomainPanicLevel` attribute (see Section 4.3.6) to promote a domain panic to a system panic, it is also good idea to run the `verify` command on the panicked domain to ensure that it is not damaged. If your filesets

were unmounted at the time of the crash, or if you have remounted them successfully and have run the `verify` command (if needed), you can mount the filesets on a different version of the operating system, if appropriate.

---

## Managing the Advanced File System with the AdvFS GUI

The AdvFS Graphical User Interface (GUI), available with AdvFS Utilities, provides a visual representation of the AdvFS file system. The GUI is designed to run under the Common Desktop Environment (CDE). You can use the GUI as a remote manager to monitor your system or to search for available volumes. The GUI does not exactly duplicate the file system management tasks available from the command line. Rather, it allows you to visualize your file system structure and perform the most common operations on volumes, domains, filesets, and clones.

To access this utility, you must register the AdvFS Utilities license and you must have root-user privilege. For information about license activation see [License Registration](#).

### 6.1 Installing the GUI

To load the GUI, choose the AdvFS Utilities subset when you update or install the Version 5.0 operating system.

Table 6-1 contains the subset titles, names (where xxx is the subset identification number), and descriptions of the subsets that make up the AdvFS GUI. The disk space requirements for loading and running AdvFS Utilities software subsets are shown in Table 6-2.

**Table 6-1: AdvFS GUI Subsets**

Subset	Name	Description
AdvFS Utilities	(OSFADVFSxxx)	Contains a set of advanced utilities licensed for managing AdvFS
AdvFS Graphical User Interface (dtadvfs)	(OSFXADVFSxxx)	Contains the AdvFS Graphical User Interface and online help files

**Table 6–1: AdvFS GUI Subsets (cont.)**

AdvFS Agent (advfsd)	(OSFADVFSDAEMONxxx)	Contains the AdvFS agent, which runs in the traditional style of the UNIX daemon
AdvFS J apanese Graphical User Interface	(OISJ PXADVFSxxx)	Localization files that must also be loaded when the J apanese version of the GUI is run

**Table 6–2: Disk Space Requirements for the AdvFS GUI**

Utility	/(root)	/var	/usr
Utilities Subset	0	0	300
GUI (dtadvfs)	30	20	7000
Agent (advfsd)	0	50	800
J apanese localization files	0	1	661

## 6.2 Components of the GUI

There are two parts to the GUI: `advfsd`, the agent, and `dtadvfs`, the actual graphical interface. The agent must be running for the GUI to operate.

### 6.2.1 GUI Agent (advfsd)

The AdvFS GUI agent issues commands and obtains system information for the GUI. The agent is automatically started at boot time and when the OSFADVFSDAEMON subset is installed. It runs unseen in the background.

Under normal conditions, `advfsd` does not need to be run manually. If you want to start or stop the agent at any time, do so from the command line (see `advfsd(8)`).

Only one agent can be running on a system at a given time. If you attempt to start a second copy of `advfsd`, it will fail.

The agent allows Simple Network Management Protocol (SNMP) clients such as NetView® to request AdvFS information. This is not a two-way path: SNMP clients cannot issue system configuration commands to `advfsd`.

If the agent is not running, the GUI cannot operate. The agent periodically asks for information from the system on which it is running. It then updates the file systems and the storage device information passed to the GUI display. It also evaluates any free space alert conditions. The time interval for when the agent scans the system disks is the agent state monitor interval.

To change the agent state monitor interval, see Section 6.3.1. The interval is saved between restarts of the agent.

## 6.2.2 GUI (dtadvfs)

Only the root user can start `dtadvfs`. It can be started in a terminal window or from a CDE icon. The file system operations that the GUI can perform are described in Section 6.4.

To start the GUI from the command line, type:

```
# /usr/bin/X11/dtadvfs &
```

To start the GUI using Common Desktop Environment (CDE) icons:

1. Click the Application Manager on the CDE toolbar.
2. Double-click the System\_Admin icon in the Application Manager window.
3. Double-click the Storage\_Management icon in the System\_Admin window.
4. Double-click the Advanced File System icon in the Storage\_Management window.

When the GUI is running, window displays are periodically updated to reflect the changing characteristics of the file system. The GUI refresh interval, the time interval between updates, is 15 minutes by default. To change the GUI refresh interval when the GUI is running, see Section 6.3.1. The GUI refresh interval is not saved between restarts of the GUI.

### 6.2.2.1 GUI Security

Each Tru64 UNIX system that the GUI will manage has two optional security files associated with it: a password file and a file of allowable hosts. The password file restricts the use of the GUI managing a particular system to administrators who know the password. The allowable hosts file allows only GUIs running on systems listed in the file to remotely manage it. These files are configured through an editor of your choice. They cannot be accessed directly from the GUI.

The root user creates the password file, `/var/advfs/daemon/socket/gui.password`, on the system for which the password is to be used. There is one password file containing one password per system. (It is not a good idea to use the root password.) The file contains the password in plain text. Only users who enter the correct password can run a GUI connected to the system. A user wishing to use the GUI to manage the system from a remote location must know this password in order to connect to the system.

The allowable hosts file, `/var/advfs/daemon/socket/hosts.allow`, is also created by the root user. It contains a plain text list of all systems (hosts) on which a GUI may operate to manage the system. The `hosts.allow` file will automatically include the GUI running locally on the system; that is, it will include itself. If you want to allow others to remotely manage your system, you must include them in your `hosts.allow` file to authorize the agent to send your file system information to their system. If you have protected your system with a password in the system's `gui.passwd` file, remote users will also have to know that password.

See `advfsd(8)` and `dtadvfs(8)` for more information.

#### 6.2.2.2 Ignoring Disks

Create the `/var/advfs/daemon/disks.ignore` file to specify a list of disks that the agent will not examine. The `disks.ignore` file contains a plain-text list of disk drives, one per line. This file is useful because performance may be reduced when there are off-line HSZ devices or spun-down disk drives.

Since the agent processes the `disks.ignore` file every time the disks are checked, disks that fail can be added to the file and disks listed in the file that become available can be removed. It is not necessary to stop the agent. See `advfsd(8)` for more information.

You cannot ignore an LSM volume by including the LSM volume name in the `disks.ignore` file; you must list the disks from which the LSM volume is built. To ignore a complete disk group, you must list all disks in it. Because all partitions on the listed disks will be ignored, unexpected results may occur if a disk has partitions belonging to more than one disk group.

#### 6.2.2.3 Log Files

Entries are generated in the agent log file, `/var/advfs/daemon/logs/advfsd`, and the GUI log file, `/var/advfs/gui/advfs_gui.log`, as processing proceeds. It is a good idea to periodically check these files. Both the agent and the GUI log files are periodically renamed to prevent unrestricted consumption of disk space.

#### 6.2.2.4 Help

The help system contains background information about the AdvFS file system as well as specific information about the GUI. Each operation is described. Detailed explanations of the dialog boxes used to perform file system activities are provided.



### 6.2.2.5 Monitoring Remote File Systems

From your system you can remotely monitor file systems that have your system listed in their `hosts.allow` file. You can attach to these systems through the GUI:

1. Choose **Host...** from the AdvFS menu of the Main window.
2. The Select a Host dialog box appears in which you can select another host system or choose to modify the host list to add or delete host systems. This list is created by you and saved on your system to identify the hosts you connect to regularly.
3. Select the host and click Connect.

If you try to connect to a host but are unsuccessful, you are no longer connected to any host so you must repeat the connection process for another system.

Note that the host list is for convenience only. It does not provide any additional security. You can always type a host name in the Selection block of the Select a Host dialog box.

### 6.2.2.6 Managing LSM Volumes

If you are using LSM volumes, it is important to know how the GUI treats these volumes:

- An LSM volume will be listed as available if it is labeled `unused`.
- **Do not** use LSM to change the size of an LSM volume that is in use by an AdvFS domain. AdvFS is unable to accommodate this change.
- If you have added the LSM volume `/dev/vol/volname` to the domain from the command line, the AdvFS GUI does not recognize that it is the same as `/dev/vol/rootdg/volname`. If the volume has mounted filesets, it will appear as unknown in the Devices and Volumes window; if it has no mounted filesets, it will appear as unavailable. If you try to add it to a domain, you will get an error message saying it is in use.
- If you encapsulate a volume in use by AdvFS into LSM while the GUI is running, the volume name will not be updated in the GUI window. You must exit and restart the GUI to display the correct name.
- You cannot ignore an LSM volume by including the LSM volume name in the `disks.ignore` file; you must list the disks from which the LSM volume is built.

## 6.3 Displaying File System Information

The objects (components) that make up the file system (volumes, domains, filesets, and fileset clones) are viewed in the GUI windows in an object tree that shows their hierarchical relationship to each other. If an object contains other objects, a folder icon appears to its left. Clicking on the folder icon expands or collapses the object tree; that is, displays or hides the objects below it in the tree. For example, clicking on a domain folder icon displays or hides the filesets that belong to that domain.

You can change your window views and look at critical information about the objects in a number of ways. You can view:

- Main window information

The first window that appears, the Main window, provides a comprehensive view of the file system. It is the starting point for all GUI-managed tasks and the primary screen for monitoring file system status. Use the items on the View menu to show the status of domains and filesets; domains and volumes; and domains, filesets, and volumes. You can also use the View menu to change from one of these views to viewing fileset quotas and back.

- Devices and volumes

Choose **Devices and Volumes** from the Configuration menu for a different detailed view. The Devices and Volumes window displays all the physical disks, disk groups, and logical volumes on the system and can be used to determine volume type or partition availability.

- Object characteristics

Choose **Show...** from the Configuration menu to display more complete information about the object that you have highlighted in the Main window display.

You can usually select **Show...** as an alternate way of performing the file system management tasks described in the following sections. For example, highlight the domain and choose **Show...** to see an object dialog box that includes the creation date and number of log pages for that domain.

Note that double-clicking a domain, fileset, clone, or volume in the object tree is the same as choosing **Show...** for that object. Double-clicking does not work for hosts, devices, or partitions or for the volume icon in the Devices and Volumes window. Single-clicking an object in an object tree only highlights it. No new information is presented.

Clicking the right mouse button on an object brings up a list of commands appropriate for that object.

If you attempt to access a menu item and find it unavailable (grayed out), check that the appropriate object is highlighted. For example, you need to select a fileset before you can create a clone for it.

### 6.3.1 Choosing Units for Displaying Information

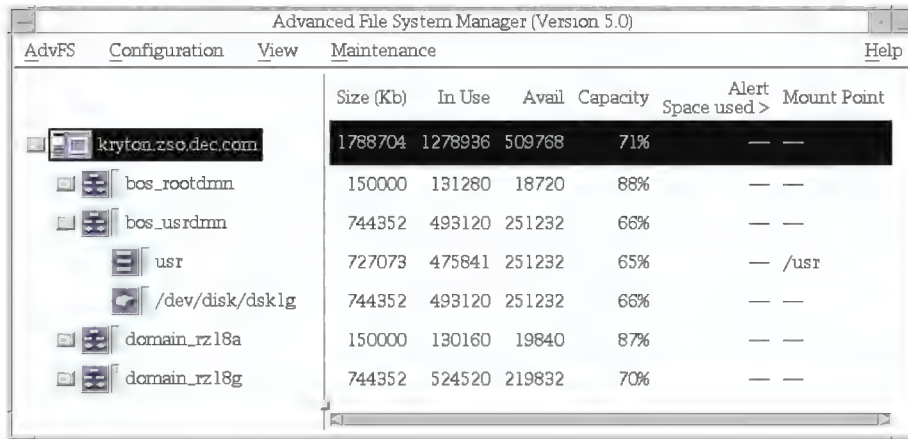
You can customize the display of your file system information by choosing **Options** from the AdvFS menu. All options except the agent state monitor interval are reset each time the GUI is restarted.

- Units  
Choose to display disk space usage in 512-kilobyte blocks, kilobytes, megabytes, gigabytes, or terabytes.
- Update Interval  
Choose the GUI refresh rate.  
Choose the Agent state monitor interval.
- Free Space Alert  
Choose between space used > and free space <  
Choose to display space available as a percent or in the units you have chosen for the display.
- Fileset quotas  
Choose to calculate quotas by percent of space or by the amount of space used.

### 6.3.2 Main Window

When the Main window, as shown in Figure 6-1, is first displayed, disk-space usage information is shown for the host and for domains. Options on the View menu let you display domains, filesets, and volumes in different combinations.

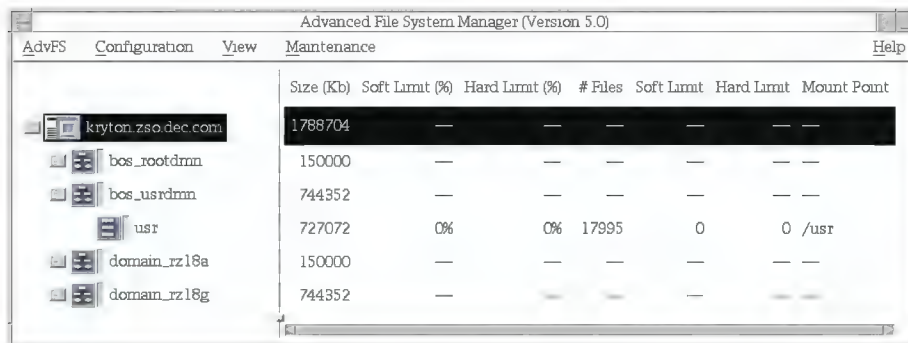
**Figure 6–1: Main Window Showing Disk Usage Information**



	Size (Kb)	In Use	Avail	Capacity	Alert Space used >	Mount Point
kryton.zso.dec.com	1788704	1278936	509768	71%	—	—
bos_rootdmn	150000	131280	18720	88%	—	—
bos_usrdmn	744352	493120	251232	66%	—	—
usr	727073	475841	251232	65%	—	/usr
/dev/disk/dsk1g	744352	493120	251232	66%	—	—
domain_rz18a	150000	130160	19840	87%	—	—
domain_rz18g	744352	524520	219832	70%	—	—

From the Main window you can also view fileset quota information, as shown in Figure 6–2, by choosing **Fileset Quotas** from the View menu. You can return to viewing domain, fileset, and volume information by selecting the items you wish to view from the same View menu.

**Figure 6–2: Main Window Showing Fileset Quota Information**



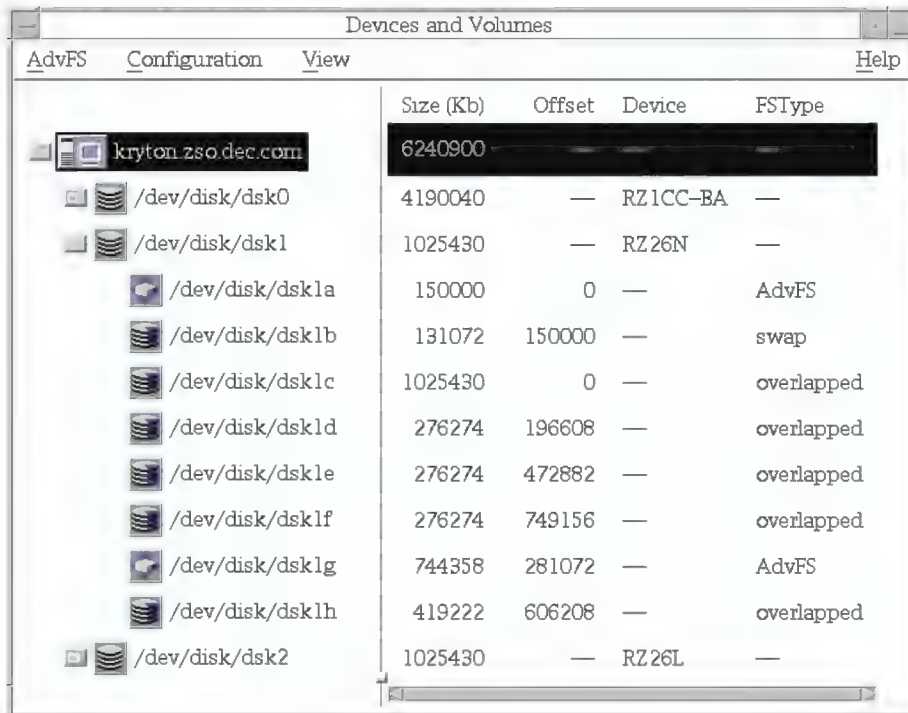
	Size (Kb)	Soft Limit (%)	Hard Limit (%)	# Files	Soft Limit	Hard Limit	Mount Point
kryton.zso.dec.com	1788704	—	—	—	—	—	—
bos_rootdmn	150000	—	—	—	—	—	—
bos_usrdmn	744352	—	—	—	—	—	—
usr	727072	0%	0%	17995	0	0	/usr
domain_rz18a	150000	—	—	—	—	—	—
domain_rz18g	744352	—	—	—	—	—	—

The Main window has menu selections to accomplish file system management tasks for domains, filesets, clones, and volumes. Menu commands call up dialog boxes to accomplish these tasks.

### 6.3.3 Devices and Volumes Window

The Devices and Volumes window, as shown in Figure 6–3, shows disk groups and logical volumes. It identifies the size and type of disks and what each partition is used for. This view is particularly useful if you wish to enlarge your domain and need to locate a volume to use. From the View menu of the Devices and Volumes window, you can sort your objects, choose to show only available volumes, or choose to show overlapped volumes.

**Figure 6–3: Devices and Volumes Window**



Size (Kb)	Offset	Device	FSType
6240900			
4190040	—	RZ1CC-BA	—
1025430	—	RZ26N	—
150000	0	—	AdvFS
131072	150000	—	swap
1025430	0	—	overlapped
276274	196608	—	overlapped
276274	472882	—	overlapped
276274	749156	—	overlapped
744358	281072	—	AdvFS
419222	606208	—	overlapped
1025430	—	RZ26L	—

## 6.4 Performing File System Operations

You can perform file system management tasks on domains, filesets, fileset clones, and volumes directly from the GUI. In general, you must select the appropriate object before you can proceed. In the sections below, menu selections are shown in bold face type. Note that domain and file domain are synonymous.

### 6.4.1 Operations on a Domain

File system management tasks for domains are initiated from the Main window by selecting **Show** or **File Domain** on the Configuration menu. In addition, from the Devices and Volumes window you can create a new domain and add volumes to an existing domain.

#### 6.4.1.1 Creating a Domain

To create a domain, you must assign a name and select an initial volume. You can set a free space alert.

If your system predates DIGITAL UNIX 4.0E, you will have the option to access **Advanced** options that modify the bitfile metadata table (BMT). This will allow you to avoid out-of-space messages that may occur for file systems that contain a very large number of files (over about 50,000), such as file systems that support Usenet news servers. (Later versions of the operating system do not have this problem and the option is not offered.)

A domain is not completely defined until you have created at least one fileset. A domain is not active unless there is a mounted fileset. To create a domain using the Main window:

1. Choose **File Domain** from the Configuration menu.
2. Choose **New** from the File Domain menu.
3. In the New File Domain dialog box, enter the domain name and highlight an available volume in the object tree. You may need to expand the device list by clicking on the folder to the left of the device name. You can also set the free space alert.

If Advanced options is available, you can direct AdvFS to grow the BMT. You can either enter the number of files you expect and have the system estimate the extent size, or you can specify the number of extent pages directly. You can set the extent size to any value, but it is suggested that for every 100,000 files you increase the BMT extent size by 256 pages.

If you are not certain what storage device to use for the new domain, work from the Devices and Volumes window (on the Configuration menu) to get a complete view of the storage on your system:

1. Choose **Devices and Volumes** from the Configuration menu of the Main window.
2. In the Devices and Volumes window, choose **Show Available Volumes Only** from the View menu.
3. Choose a volume that is labeled as available.
4. Choose **New File Domain** from the Configuration menu.
5. In the New File Domain dialog box, create the domain and set the free space alert. If you are monitoring a file system prior to Version 4.0E, see the Main window instructions above for directions on using the Advanced button.

Creating a domain automatically generates an entry in the `/etc/fdmns` directory.

#### 6.4.1.2 Setting a Free Space Alert for a Domain

Set a free space alert to inform you when the free space threshold in the domain is reached or passed. When you set the alert, you can choose to automatically run a script when the alert threshold is crossed. See `/usr/advfs/daemon/scripts` for examples of scripts used by the agent to execute commands.

To set the free space alert from the Main window:

1. In the object tree, highlight the domain.
2. Choose **File Domain** from the Configuration menu.
3. Choose **Set free space alert** from the File Domain menu.
4. In the File Domain dialog box, set the free space alert.

You must reset the free space alert when the size of a domain has changed. Free space alert values are evaluated using actual size (for example, kilobytes), not percentage of space.

#### 6.4.1.3 Enlarging a Domain

Add volumes to transform a single-volume domain (except the root domain, which can only have one volume) into a multivolume domain or to enlarge a multivolume domain (see Section 1.4.6).

To add a volume to a domain:

1. In the object tree, highlight the domain.
2. Choose **File Domain** from the Configuration menu.
3. Choose **Add Volume** from the File Domain menu.
4. In the Add Volume dialog box, highlight an available volume to add. You may need to expand the view of the volumes by clicking the folder icon to the left of the device name.

If you do not know which volume to add to your domain, work from the Devices and Volumes window to get a complete view of the storage on your system:

1. Choose **Devices and Volumes** from the Configuration menu of the Main window.
2. In the Devices and Volumes window, choose **Show available volumes only** from the View menu.
3. Click a volume that is labeled available.
4. Choose **Add to domain** from the Configuration menu.

5. In the Add Volume dialog box, choose the domain to which the new volume will be added.

It is a good idea to balance your domain after you have added a volume. This will distribute existing files to the new volume.

#### 6.4.1.4 Removing a Domain

You can delete a domain only after all filesets and clone filesets in the domain are unmounted. You will get an error message if you try to remove a domain with mounted filesets. When you remove a domain as a method of deleting all filesets, its entry in the `/etc/fdmns` directory is removed. However, this may present a security hole since only pointers are changed and no data on the volume is removed. It may be possible to access the fileset data from the command line with the `salvage` utility (see Section 5.4.6). If you need the increased security, remove each fileset individually.

The deleted domain name remains in the `/etc/fstab` file unless you modify the file by choosing the **Modify /etc/fstab** option in the dialog box or change it from the command line. If you do not update this file, you will get error messages when you reboot the system or when you enter a `mount -a` command from the command line. If you do not delete the file name from the `/etc/fstab` file, you can do so at a later time from the command line.

To remove a domain:

1. In the object tree, highlight the domain.
2. Choose **File Domain** from the Configuration menu.
3. Choose **Delete** from the File Domain menu.
4. In the Delete File Domain dialog box, delete the domain and back up and modify the `/etc/fstab` file.

---

#### Note

---

When you unmount your filesets and clone filesets, you must choose to modify the `/etc/fstab` file. If you do not do so at that time, you must manually edit the `/etc/fstab` file. Once you delete the domain you no longer have access to the filesets.

---

#### 6.4.1.5 Renaming a Domain

You can assign a new name to an existing domain (see Section 1.4.9). The old name remains in the `/etc/fstab` file unless you modify the file by choosing the **Modify /etc/fstab** option in the dialog box or change it from the command line. If you do not update the `/etc/fstab` file, the filesets in this domain



will not mount when you reboot the system or when you enter the `mount -a` command from the command line. If you must update the `/etc/fstab` file at a later time, do so from the command line.

Unmount all filesets before renaming the domain. An error will occur if you try to rename a domain with mounted filesets. You cannot rename a domain with the name of an existing domain.

To rename a domain:

1. In the object tree, highlight the domain.
2. Choose **File Domain** from the Configuration menu.
3. Choose **Rename** from the File Domain menu.
4. In the Rename File Domain dialog box, enter the new name and back up and modify the `/etc/fstab` file.

## 6.4.2 Operations on a Fileset

File system management tasks for filesets are initiated from the Main window by selecting **Show...** or **Fileset** from the Configuration menu once you have highlighted the fileset on which you wish to operate. It is important that you choose to modify and back up the `/etc/fstab` file as you perform fileset operations. If you do not, there is a chance that the `/etc/fstab` file will not be correct for subsequent operations. If you must update the `/etc/fstab` file at a later time, do so from the command line.

### 6.4.2.1 Creating a Fileset

You cannot create a fileset until you have a domain with which to associate it (see Section 6.4.1.1).

To create a fileset:

1. In the object tree, highlight the domain for which the fileset will be created.
2. Choose **Fileset** from the Configuration menu.
3. Choose **New** from the Fileset menu.
4. In the New Fileset dialog box, create the fileset, mount the fileset, back up and modify the `/etc/fstab` file, set a free space alert, and set fileset quotas (as desired).

Note that when you create a fileset, you can set the free space alert only as a percentage of the available space.

#### 6.4.2.2 Mounting a Fileset

When you mount a fileset, you make its files available. The display in the Main window indicates which filesets are mounted. A domain is active when at least one fileset is mounted.

To mount a fileset:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Mount** from the Fileset menu.
4. In the Mount Fileset dialog box, create a mount point if it does not exist and back up and modify the `/etc/fstab` file.

#### 6.4.2.3 Unmounting a Fileset

When you unmount a fileset, its files are no longer available. You must unmount the fileset before you can remove it or remove the domain to which it belongs. The display in the Main window indicates whether filesets are mounted or not mounted.

To unmount a fileset:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Unmount** from the Fileset menu.
4. In the Unmount Fileset dialog box, unmount the fileset and back up and modify the `/etc/fstab` file.

#### 6.4.2.4 Setting a Free Space Alert for a Fileset

Set a free space alert to warn when the free space threshold in the fileset is reached or passed. You must mount the fileset before you can set the alert. After you set the alert, you can choose to automatically run a script when the alert threshold is crossed.

To set the free space alert:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Set free space alert** from the Fileset menu.
4. In the Fileset dialog box, set the free space alert.

You must reset the free space alert when the size of a fileset has changed. Free space alert values are evaluated using actual size (for example, kilobytes), not percentage of space.

#### 6.4.2.5 Setting Fileset Quotas

Set fileset quotas to limit the amount of space the fileset can consume. If you do not set quotas, any fileset can use all the available space in the domain. You can only set quotas on mounted filesets.

To set fileset quotas:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Set fileset quotas** from the Fileset menu.
4. In the Fileset dialog box, set the fileset quotas.

You can view fileset quota information from the Main window by choosing **Fileset Quotas** from the View menu. To return to viewing disk usage, choose the items you wish to view from the View menu.

#### 6.4.2.6 Removing a Fileset

You must unmount a fileset before you can delete it. You can do this as part of the removal process. You cannot remove a fileset that has a clone. You must remove the clone first. Removing a fileset removes all files in that fileset.

To remove a fileset:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Delete** from the Fileset menu.
4. In the Delete Fileset dialog box, unmount the fileset, delete it, and back up and modify the `/etc/fstab` file.

If all the filesets are unmounted, the fastest way to remove all filesets is to remove the domain to which they belong. However, this may present a security hole because it may be possible to access the data with the `salvage` utility (see Section 5.4.6).

#### 6.4.2.7 Renaming a Fileset

You must unmount a fileset before you can rename it. You can do this as part of the renaming process. The new fileset name must be unique within the domain.

To rename a fileset:

1. In the object tree, highlight the fileset.
2. Choose **Fileset** from the Configuration menu.
3. Choose **Rename** from the Fileset menu.
4. In the Rename Fileset dialog box, unmount the fileset, enter a new name, and back up and modify the `/etc/fstab` file.

---

**Note**

---

If you rename a fileset that has never been mounted (does not already have an `/etc/fstab` file entry) and check the modify `/etc/fstab` box, you do not get a new entry for the fileset name. You must manually edit the `/etc/fstab` file to add the new entry.

---

If you rename a fileset, its clone can no longer track it. You must delete the old clone and create a new one because clones cannot be renamed.

### 6.4.3 Operations on a Clone Fileset

When you operate on a clone, it is important to modify and back up the `/etc/fstab` file as you perform clone operations. If you do not, there is a chance that the `/etc/fstab` file will not be correct in subsequent operations. If you must update the `/etc/fstab` at a later time, do so from the command line.

#### 6.4.3.1 Creating a Clone Fileset

Creating a clone fileset allows you to back up files while the file system is on line (see Section 3.2). You can create only one clone for a fileset.

To create a clone:

1. In the object tree, highlight the fileset that you want to clone.
2. Choose **Clone** from the Configuration menu.
3. Choose **New** from the Clone menu.
4. In the New Clone dialog box, enter a clone name, create a mount point, and back up and modify the `/etc/fstab` file.

#### 6.4.3.2 Mounting a Clone

You must mount a clone in order to access it. An unmounted clone tracks changes to a fileset but cannot be read. Clones are mounted as read-only.

To mount a clone:

1. In the object tree, highlight the clone.
2. Choose **Clone** from the Configuration menu.
3. Choose **Mount** from the Clone menu.
4. In the Mount Clone dialog box, create a mount point if it does not exist and back up and modify the `/etc/fstab` file.

#### 6.4.3.3 Unmounting a Clone

You cannot access an unmounted clone, but it still tracks fileset changes.

To unmount a clone:

1. In the object tree, highlight the clone.
2. Choose **Clone** from the Configuration menu.
3. Choose **Unmount** from the Clone menu.
4. In the Unmount Clone dialog box, unmount the clone and back up and modify the `/etc/fstab` file.

#### 6.4.3.4 Removing a Clone

You must unmount a clone before you can remove it. You can do this as part of the removal process.

To remove a clone:

1. In the object tree, highlight the clone.
2. Choose **Clone** from the Configuration menu.
3. Choose **Delete** from the Clone menu.
4. In the Delete Clone dialog box, unmount the clone, delete it, and back up and modify the `/etc/fstab` file.

### 6.4.4 Operations on a Volume

File system management tasks for volumes are initiated from the Main window by selecting **Show** or **Volume** from the Configuration menu once you have highlighted the volume on which you wish to operate. (In order to view volumes, the View menu must be set to **Domains and volumes** or **Domains, filesets, and volumes**.)

#### 6.4.4.1 Setting a Free Space Alert for a Volume

Set a free space alert to inform you when the free space threshold in the volume is reached or passed.

To set the free space alert:

1. In the object tree, highlight the volume.
2. Choose **Volume** from the Configuration menu.
3. Choose **Set free space alert** from the Volume menu.
4. In the Volume dialog box, set the free space alert.

When you set the alert, you can choose to automatically run a script when the threshold is crossed.

#### 6.4.4.2 Adding a Volume

Adding a volume is the same as enlarging a domain (see Section 6.4.1.3). See Section 1.3 for detailed information about volumes.

#### 6.4.4.3 Removing a Volume

You can remove a volume from a domain at any time without interrupting the logical structure of the filesets in the domain (see Section 1.4.7). The data that was stored on that volume will be moved to other volumes in the domain. In order to complete successfully, there must be room on the remaining volumes.

Before you can remove a volume from a domain, all filesets in that domain must be mounted. You cannot remove a volume while you are balancing or defragmenting a domain associated with that volume.

To remove a volume:

1. In the object tree, highlight the volume.
2. Choose **Volume** from the Configuration menu.
3. Choose **Remove** from the Volume menu.
4. In the Remove Volume dialog box, confirm the removal.

#### 6.4.5 Maintaining AdvFS

File system maintenance tasks are initiated from the Main window through the Maintenance menu. The maintenance tools improve read/write performance by altering the way files are mapped on the disk. They can be run while the system is on line, and their operation is transparent to system

users and to applications. The maintenance tasks can be stopped without harm to the domain. The activity that has already taken place will remain.

#### 6.4.5.1 Balancing a Multivolume Domain

You can use the GUI to initiate the balance process to evenly distribute files among volumes (see Section 4.3.2). You cannot balance a domain while you are defragmenting, adding or removing volumes, or removing filesets on the same domain. You can choose to abort the operation at any time. The domain you want to balance must have all filesets mounted. If you try to balance a domain that includes unmounted filesets, you will get an error message.

To balance a domain:

1. In the object tree, highlight the domain.
2. Choose **Balance** from the Maintenance menu.
3. In the Balance dialog box, start the balance process.

#### 6.4.5.2 Defragmenting a Domain

You can make files in a domain more contiguous by defragmenting the domain. Defragmenting also consolidates free space so files created later are also less fragmented (see Section 4.3.1). You cannot defragment a domain while you are balancing, adding or removing volumes, or removing filesets on the same domain. You can choose to abort the defragment operation at any time. The domain you want to defragment must have all filesets mounted. If you try to defragment a domain that includes unmounted filesets, you will get an error message.

To defragment a domain:

1. In the object tree, highlight the domain.
2. Choose **Defragment** from the Maintenance menu.
3. In the Defragment dialog box, start the defragment process and specify how long it is to operate.

### 6.5 Troubleshooting GUI Operation

Table 6-3 presents some GUI problems and solutions. Check `dtadvfs(8)` and `advfsd(8)` for additional information.

**Table 6–3: GUI Troubleshooting**

Problem	Cause/Solution
GUI starts slowly	Mount at least one fileset from the command line.
Advanced File System icon not in the Application Manager - Storage Management window	Installation is not complete.
Permission denied message when entering <code>dtadvfs</code> from the command line	You are not the root user.
Password Error dialog when starting <code>dtadvfs</code> from CDE icon	You have entered an incorrect root password. Do not reenter. Cancel the dialog and try again.
Object key is invalid or Can't get the object data error message	GUI sometimes cannot track several actions simultaneously. Restart the GUI.
Slow performance or unexplained error messages appear	<p>Network overload (the GUI cannot get configuration data in a reasonable number of attempts) or system overload (the agent gets few or no CPU cycles).</p> <p>If there are several off-line HSZ devices, unmounted filesets, or spun down disk devices, add them to the <code>disks.ignore</code> file.</p> <p>If none of the above appears to be the problem, the agent is probably hung. Exit the GUI and stop the agent:</p> <pre># /sbin/init.d/advfsd stop</pre> <p>Then restart the agent and the GUI:</p> <pre># /sbin/init.d/advfsd start # /usr/bin/X11/dtadvfs</pre>
Incorrect error message	Configuration tasks are running simultaneously and more than one task failed.
Free space alert value incorrect	Reset. Alert values are evaluated using actual size (for example, in kilobytes), not percentage of space.
Domain deleted but associated volumes shown as in use	Manually change the disk label on the device; use the command line <code>disklabel</code> command or the CDE Disk Configuration utility.
Volume removed but shown	Manually change the disklabel on the device; use the command line <code>disklabel</code> command or the CDE Disk Configuration utility.
<code>advfsd</code> consuming high CPU and I/O resources	Change the agent state monitor interval to reduce the polling frequency (see Section 6.2.1).



**Table 6–3: GUI Troubleshooting (cont.)**

	<p>Decrease the number of LSM disk groups and volumes.</p> <p>Decrease the number of unmounted filesets. Inactive domains are particularly slow. Use the <code>disks.ignore</code> file (see Section 6.2.2.2 ).</p> <p>If none of the above appears to be the problem the agent is probably hung. Exit the GUI and stop the agent:</p> <pre># /sbin/init.d/advfsd stop</pre> <p>Then restart the agent and the GUI:</p> <pre># /sbin/init.d/advfsd start # /usr/bin/X11/dtadvfs</pre>
System panics at boot time	<p>Check for a bad domain. Panic will occur even if the domain has been removed from the <code>/etc/fstab</code> file. Do one of the following in single-user mode, then reboot to multiuser mode:</p> <ul style="list-style-type: none"><li>- Repair the domain.</li><li>- Move the bad domain from the <code>/etc/fdmns</code> directory to another directory. (This will make repair difficult.)</li></ul>
Sort gives incorrect results	<p>Check that no data fields to be sorted are marked unavailable. Mount filesets if needed.</p>
Volume in use message, when adding a volume	<p>Check that this is not an unrecognized LSM volume (see Section 6.2.2.6).</p>
X or Motif® problems	<p>Failures related to these can be ignored. Full keyboard support as defined in the Motif style guide is not implemented.</p>

---



# A

---

## Accessing File System Information with the SysMan Menu

You must be root to use the SysMan Menu. You can open the menu:

- From the command line enter:  

```
# /usr/sbin/sysman
```
- From the Common Desktop Environment (CDE) front panel if your system is running in a graphics environment with CDE:
  1. Select the SysMan Application from the front panel.
  2. Select SysMan Menu.

For AdvFS tasks select the “Storage” management task then choose “File System Management Utilities.” “Advanced File System (AdvFS) Utilities” lists the AdvFS-related tasks.

See `sysman(8)` for more information.



# B

## AdvFS Commands

This appendix summarizes the AdvFS commands. The reference pages provide detailed information.

### B.1 AdvFS Base System Commands

The following tables list and describe each of the AdvFS commands available in the base portion of AdvFS. These commands are included with the basic license; they do not require a layered product license. If you installed the reference page subset, you can access reference pages for each of these commands by issuing the `man` command. Commands marked with an asterisk (\*) are functions for which the AdvFS Graphical User Interface (GUI) has equivalent capability.

**Table B–1: AdvFS Configuration Commands**

Command	Description
<code>chfile</code>	Changes the attributes of a file
<code>chvol</code>	Changes the attributes of a volume
<code>defragment*</code>	Makes the files in a domain more contiguous
<code>mkfdmn*</code>	Creates a domain
<code>mkfset*</code>	Creates a fileset within a domain
<code>renamefset*</code>	Renames an existing fileset
<code>rmfdmn*</code>	Removes a domain
<code>rmfset*</code>	Removes a fileset from a domain
<code>switchlog</code>	Moves the AdvFS log file to a different volume in a domain

**Table B–2: AdvFS Information Display Commands**

Command	Description
<code>advfsstat*</code>	Displays file system statistics
<code>ncheck</code>	Displays the tag and full path name for each file in the file system
<code>showfdmn*</code>	Displays the attributes of a domain
<code>showfile</code>	Displays the attributes of a file

**Table B–2: AdvFS Information Display Commands (cont.)**

<code>showfsets*</code>	Displays the attributes of filesets in a domain
<code>vdf*</code>	Displays used and available disk space for a fileset or a domain

**Table B–3: AdvFS Backup Commands**

Command	Description
<code>rvdump</code>	Remotely performs full and incremental fileset backup
<code>rvrestore</code>	Remotely restores files from backup media
<code>vdump</code>	Performs full and incremental fileset backup
<code>vrestore</code>	Restores files from backup media

**Table B–4: AdvFS Check and Repair Commands**

Command	Description
<code>advscan</code>	Locates AdvFS partitions on disks
<code>mountlist</code>	Checks for mounted AdvFS filesets
<code>tag2name</code>	Prints the path name of a file given the tag number
<code>salvage</code>	Recovers file data from damaged AdvFS domains
<code>verify</code>	Checks for and repairs file system inconsistencies

**Table B–5: AdvFS Quota Commands**

Command	Description
<code>chfsets</code>	Changes file and block quotas
<code>edquota</code>	Edits user and group quotas
<code>quot</code>	Summarizes file and block information
<code>quota</code>	Displays disk usage and limits by user or group
<code>quotacheck</code>	Checks file system quota consistency
<code>quotaoff</code>	Turns quotas off
<code>quotaon</code>	Turns quotas on
<code>repquota*</code>	Summarizes quotas for a file system

**Table B–6: AdvFS Disk Structure Commands**

Command	Description
<code>nvbmtpg</code>	Displays a formatted page of the bitfile metadata table (BMT)
<code>nvfragpg</code>	Displays file fragment information
<code>nvlogpg</code>	Displays a formatted page of the log
<code>nvtagpg</code>	Displays a formatted page of the tag directory
<code>savemeta</code>	Captures a domain's metadata
<code>shblk</code>	Displays unformatted disk blocks
<code>shfragbf</code>	Displays file fragment information
<code>vfilepg</code>	Displays a page of an AdvFS file
<code>vsbmpg</code>	Displays a page of the storage bitmap

## B.2 AdvFS Utilities Commands

The following table lists and describes AdvFS Utilities commands. These commands require the optional AdvFS Utilities product license. If you installed the AdvFS Utilities reference page subset, you can access reference pages for each of these commands by entering the `man` command. The commands marked with an asterisk (\*) are functions for which the AdvFS GUI has equivalent capability.

**Table B–7: AdvFS Utilities Commands**

Command	Description
<code>addvol*</code>	Adds a volume to an existing domain
<code>advfsd</code>	Starts the AdvFS GUI agent (daemon)
<code>balance*</code>	Balances the percentage of used space between volumes
<code>clonefset*</code>	Creates a read-only copy of a fileset
<code>dtadvfs</code>	Starts the AdvFS GUI
<code>migrate</code>	Moves a file to another volume in the domain
<code>mktrashcan</code>	Attaches directories to a trashcan directory, which stores deleted files
<code>rmtrashcan</code>	Detaches a specified directory from a trashcan directory
<code>rmvol*</code>	Removes a volume from an existing domain

**Table B–7: AdvFS Utilities Commands (cont.)**

<code>shtrashcan</code>	Shows the trashcan directory, if any, that is attached to a specified directory
<code>stripe</code>	Interleaves storage allocation of a file across two or more volumes within a domain

## B.3 Comparison of AdvFS and UFS Commands

The following table lists the AdvFS commands and equivalent or similar commands available in UFS file systems. Note that there is not a one-to-one correspondence between all AdvFS and all UFS commands.

**Table B–8: Comparison of AdvFS and UFS Commands**

AdvFS Command	UFS Command
<code>addvol</code>	None; UFS does not use volumes.
<code>advfsd</code>	None; no UFS GUI.
<code>advfsstat</code>	Use <code>iostat</code> or <code>stern</code> monitoring tools such as <code>sys_check</code> and <code>Collect</code> for Tru64 UNIX. Use event monitoring tools to watch for file system events.
<code>advscan</code>	None; UFS does not use volumes.
<code>balance</code>	None; UFS does not use volumes.
<code>chfile</code>	None; the only attributes that can be changed are ownership and protection.
<code>chfsets</code>	None; the only attributes that can be changed are ownership and protection.
<code>chvol</code>	None; UFS does not use volumes.
<code>clonefset</code>	None.
<code>defragment</code>	None.
<code>dtadvfs</code>	None; no UFS GUI.
<code>migrate</code>	Use <code>cp</code> and <code>mv</code> .
<code>mkfdmn</code> , <code>mkfset</code>	Use <code>newfs</code> .
<code>mktrashcan</code> , <code>rmtrashcan</code> , <code>shtrashcan</code>	Use CDE windowing environment, which provides a deferred deletion.
<code>mountlist</code>	Use <code>mount</code> .
<code>ncheck</code>	Same as AdvFS.



**Table B–8: Comparison of AdvFS and UFS Commands (cont.)**

nvbmptg, nvfragpg, nvlogpg, nvtagpg, savemeta, shblk, shfragbf, vfilepg, vsbmpg	None; no analogous UFS commands.
quota, edquota, quot, quotacheck, quotaoff, quotaon, repquota.	Same as AdvFS.
renamefset	None.
rmfdmn, rmfset	Unmount the file system using <code>mount</code> , remove the entry from the <code>/etc/fstab</code> file.
rmvol	None; UFS does not use volumes.
vdump, rvdump, vrestore, rvrestore	Use <code>dump</code> , <code>rdump</code> , <code>restore</code> , and <code>rrestore</code> . Can use AdvFS commands.
salvage	None.
showfdmn	None.
showfile	Use <code>file</code> and <code>ls -l</code> .
stripe	None.
switchlog	None.
tag2name	None.
vdf	Use <code>df</code> and <code>du</code> .
verify	Use <code>fsck</code> and <code>bcheckrc</code> .

---



# C

---

## Converting File Systems

This appendix contains procedures to convert a `/usr` file system, the root file system, and a data file system to AdvFS. Also included are instructions for converting your entire system from AdvFS to UFS.

The methods provided here are guidelines; that is, they are suggestions that illustrate the process of conversion. Specific file names, tape drives, and disk partitions depend on your system.

The `vdump` and `vrestore` file formats are compatible for Version 4 and Version 5.0 operating systems. If you upgrade to Version 5.0, recreate the domains, and restore the data from backup (see Chapter 3), your AdvFS filesets and domains will have the updated structure (see Section 1.4.3).

### C.1 Converting a `/usr` File System to AdvFS

During the initial installation of AdvFS, you can install `/usr` on AdvFS. Converting the `/usr` (UFS) file system to AdvFS reduces the amount of time your system is down after a system failure. If you have not installed `/usr` on AdvFS, you can do so with a backup tape, an intermediate file, or from a second disk.

#### C.1.1 Using a Backup Tape

You can convert the `/usr` (UFS) file system to an equivalent AdvFS file system by backing up the existing file system to tape and restoring it to an AdvFS environment.

The following are required:

- Root user privilege
- Backup device and media
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `/usr`  
Disk partition `/dev/disk/dsk3g`
- New AdvFS configuration:  
File system `/usr`  
Disk partition `/dev/disk/dsk3g`  
Domain `usr_domain`  
Fileset `usr`

Use the following procedure as a guide for converting the file system:

1. Log in as root on the system containing the `/usr` file system.
2. Use the AdvFS `vdump` command to back up the `/usr` file system to `/dev/tape/tape0`:

```
# mt rewind
# cd /usr
# vdump -0 .
```
3. Edit the `/etc/fstab` file.
  - a. Search for the entry that mounts `/usr` as a UFS file system, such as:

```
/dev/disk/dsk3g /usr ufs rw 1 2
```
  - b. Replace it with one that mounts `/usr` as an AdvFS file system:

```
usr_domain#usr /usr advfs rw 1 0
```
4. Shut down the system:

```
# shutdown -h now
```
5. Reboot the system in single-user mode. See System Administration for instructions on invoking single-user mode.
6. In single-user mode, mount the root file system as `rw`. Create the `usr_domain` domain, and create the `usr` fileset. Use the `-F` option with the `mkfdmn` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

```
# mount -u /
# mkfdmn -F /dev/disk/dsk3g usr_domain
# mkfset usr_domain usr
```
7. Mount the `usr` fileset on the `/usr` directory:

```
# mount -t advfs usr_domain#usr /usr
```

8. Restore the `/usr` file system from tape to the `usr` fileset:

```
# vrestore -x -D /usr
```

9. Boot the system to multiuser mode. When the system prompt returns, the converted `/usr` file system is ready to use.

### C.1.2 Using an Intermediate File

You can convert the `/usr` (UFS) file system to the equivalent AdvFS file system by backing up the existing file system to a file and restoring it to an AdvFS environment.

The following are required:

- Root user privilege
- Disk space (on a different file system) for an intermediate file. (The file system containing the intermediate file can be on the same disk or a different disk.)
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `/usr`  
Disk partition `/dev/disk/dsk3g`  
Intermediate file `/tmp/usr_bck`
- New AdvFS configuration:  
File system `/usr`  
Disk partition `/dev/disk/dsk3g`  
Domain `usr_domain`  
Fileset `usr`

Use the following procedure as a guide for converting the `/usr` file system:

1. Log in as root on the system containing the `/usr` file system.
2. Use the AdvFS `vdump` command to back up the `/usr` file system to `/tmp/usr_bck`, the intermediate file:

```
# cd /usr  
# vdump -0f /tmp/usr_bck /usr
```

3. Edit the `/etc/fstab` file.
  - a. Search for the entry that mounts `/usr` as a UFS file system:
 

```
/dev/disk/dsk3g /usr ufs rw 1 2
```
  - b. Replace it with one that mounts `/usr` as an AdvFS file system:
 

```
usr_domain#usr /usr advfs rw 1 0
```
4. Shut down the system:
 

```
# shutdown -h now
```
5. Reboot the system in single-user mode. See System Administration for instructions on invoking single-user mode.
6. In single-user mode, mount the root file system as `rw`, create the `usr_domain` domain, and create the `usr` fileset. Use the `-F` option with the `mkfdmn` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.
 

```
# mount -u /
# mkfdmn -F /dev/disk/dsk3g usr_domain
# mkfset usr_domain usr
```
7. Mount the `usr` fileset on the `/usr` directory:
 

```
# mount -t advfs usr_domain#usr /usr
```
8. Restore the `/usr` file system from the intermediate file to the `usr` fileset:
 

```
# vrestore -xf /tmp/usr_bck -D /usr
```
9. Boot the system to multiuser mode. When the system prompt returns, the converted `/usr` file system is ready to use.

### C.1.3 Using a Second Disk

You can convert the `/usr` (UFS) file system on one disk to the equivalent `/usr` (AdvFS) file system on a different target disk.

The following are required:

- Root user privilege
- A second disk labeled `unused` with 5% more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:
  - File system `/usr`
  - Disk partition `/dev/disk/dsk3g`

- New AdvFS configuration:

File system `/usr`

Disk partition `/dev/disk/dsk2c`

Mount directory `/usr.advfs`

Domain `usr_domain`

Fileset `usr`

Use the following procedure as a guide for converting the `/usr` file system:

1. Log in as root on the system containing the `/usr` file system.
2. Create a domain and fileset:
 

```
# mkfdmn /dev/disk/dsk2c usr_domain
# mkfset usr_domain usr
```
3. Create a mount-point directory and mount the new fileset on the directory:
 

```
# mkdir /usr.advfs
# mount -t advfs usr_domain#usr /usr.advfs
```
4. Change to the `/usr` directory:
 

```
# cd /usr
```
5. Be certain there is no activity on the system. (You can do this by bringing the system to single-user mode.) Copy the contents of the UFS file system to the AdvFS file system:
 

```
# vdump -0f - -D . | vrestore -xf - -D /usr.advfs
```
6. Edit the `/etc/fstab` file.
  - a. Search for the entry that mounts `/usr` as a UFS file system, such as:
 

```
/dev/disk/dsk3g /usr ufs rw 1 2
```
  - b. Replace it with one that mounts `/usr` as an AdvFS file system:
 

```
usr_domain#usr /usr advfs rw 1 0
```
7. Remove the temporary directory.
 

```
# umount /usr.advfs
# rmdir /usr.advfs
```
8. Shut down and reboot the system. When the system prompt returns, the converted `/usr` file system is ready to use.

## C.2 Converting the root File System to AdvFS

By converting the root file system to AdvFS, you can boot your system from an AdvFS domain and use AdvFS as the root (/) file system. The AdvFS root domain must reside on a single disk. During initial installation you can install root on AdvFS. If you do not, you can use the following method.

---

### Note

---

Before you begin the conversion, check the size of the existing UFS root partition. The target AdvFS root domain can contain only one volume and must be large enough to accommodate the converted root file system.

---

You must have root user privilege, a second bootable disk (use partition a or c) and AdvFS installed on your system.

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `root`  
Mount directory `/newroot`  
Disk partition `/dev/disk/dsk1a`
- New AdvFS configuration:  
File system `root`  
Mount directory `/newroot`  
Disk partition `/dev/disk/dsk2a`  
Domain `root_domain`  
Fileset `root`

Use the following procedure as a guide for converting the root file system. This example assumes a device here called `<disk>`.

1. Log in as root on the system containing the root file system.
2. Create a domain and fileset:

```
# mkfdmn -r /dev/disk/dsk2a root_domain
# mkfset root_domain root
```

3. Create a mount-point directory and mount the new fileset on the directory:

```
# mkdir /newroot
# mount -t advfs root_domain#root /newroot
```



4. Be certain there is no activity on the system. (You can do this by bringing the system to single-user mode.) Restore the UFS root file system to the root fileset:

```
# vdump 0f - / | (cd /newroot; vrestore -xf -)
```

5. Make the disk with the root domain a bootable disk:

```
# disklabel -r /dev/rdisk/dsk2a > /tmp/dsk2label  
# disklabel -t advfs -r -R /dev/rdisk/dsk2a \  
/tmp/dsk2label <disk>
```

6. Edit the `/etc/fstab` file on the AdvFS root fileset to indicate the new root entry.

- a. Search `/newroot/etc/fstab` for the entry that mounts root as a UFS file system, such as:

```
/dev/disk/dsk1a / ufs rw 1 1
```

- b. Replace it with one that mounts root as an AdvFS file system:

```
root_domain#root / advfs rw 1 0
```

7. After editing is complete, shut down the system:

```
# shutdown -h now
```

8. Reset the boot default device, `BOOTDEF_DEV`, to point to the disk with the new root domain. This procedure is hardware-specific. Refer to your hardware manual for instructions.
9. Reboot the system to enable the AdvFS root file system.

The converted root file system is ready to use.

Because the AdvFS root domain is limited to one disk, you cannot use the `addvol` command to extend the root domain.

## C.3 Converting a Data File System to AdvFS

By converting your data file systems to AdvFS, you can eliminate lengthy reboots. Moreover, you can easily modify your file system configurations to meet changing system requirements.

To convert data file systems from UFS to AdvFS, you need a backup tape, an intermediate file, a second disk, or a second system.

### C.3.1 Using a Backup Tape

You can convert a data (UFS) file system to the equivalent data (AdvFS) file system by backing up the existing file system to tape with the `vdump`

command and restoring it with the `vrestore` command to an AdvFS environment.

The following are required:

- Root user privilege
- Backup device and media
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `/staff2`  
Mount directory `/staff2`  
Disk partition `/dev/disk/dsk2c`
- New AdvFS configuration:  
File system `/staff2`  
Disk partition `/dev/disk/dsk2c`  
Domain `staff_domain`  
Fileset `staff2`

Use the following procedure as a guide for converting the `/staff2` file system:

1. Log in as root on the system containing the `/staff2` file system.
2. Use the AdvFS `vdump` command to back up the `/staff2` file system to `/dev/tape/tape0`, the default tape drive:

```
# mt rewind
# mount /staff2
# vdump -0f /dev/tape/tape0 /staff2
# umount /staff2
```
3. Create the `staff_domain` domain and the `staff2` fileset. Use the `-F` option with the `mkfdmn` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

```
# mkfdmn -F /dev/disk/dsk2c staff_domain
# mkfset staff_domain staff2
```
4. Mount the new fileset on the directory:

```
# mount -t advfs staff_domain#staff2 /staff2
```
5. Restore the `/staff2` file system from tape to the `staff2` fileset:

```
# vrestore -xvf /dev/tape/tape0 -D /staff2
```

6. Edit the `/etc/fstab` file.

a. Search for the entry that mounted `/staff2` as a UFS file system:

```
/dev/disk/dsk2c /staff2 ufs rw 1 2
```

b. Replace it with one that mounts `/staff2` as an AdvFS file system:

```
staff_domain#staff2 /staff2 advfs rw 1 0
```

The converted `/staff2` file system is ready to use.

### C.3.2 Using an Intermediate File

You can convert a data (UFS) file system to the equivalent data (AdvFS) file system by backing up the existing file system to a file and restoring it to an AdvFS environment.

The following are required:

- Root user privilege.
- Disk space (on a different file system) for an intermediate file. (The file system containing the intermediate file can be on the same disk or a different disk.)
- Five percent more disk space for the converted file system.
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `/staff2`  
Disk partition `/dev/disk/dsk3g`  
Intermediate file `/tmp/staff_bck`
- New AdvFS configuration:  
File system `/staff2`  
Disk partition `/dev/disk/dsk3g`  
Domain `staff_domain`  
Fileset `staff2`

Use the following procedure as a guide for converting the `/staff2` file system:

1. Log in as root on the system containing the `/usr` file system.
2. Use the AdvFS `vdump` command to back up the `/staff2` file system to `/tmp/staff_bck`, the intermediate file:

```
# vdump -0f /tmp/staff_bck /staff2
```

3. Create the `staff_domain` domain and the `staff2` fileset. Use the `-F` option with the `mkfdomn` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

```
# mkfdomn -F /dev/disk/dsk3g staff_domain
# mkfset staff2
```

4. Edit the `/etc/fstab` file:

- a. Search for the entry that mounts `/staff2` as a UFS file system:

```
/dev/disk/dsk3g /staff2 ufs rw 1 2
```

- b. Replace it with one that mounts `/staff2` as an AdvFS file system:

```
/staff_domain#staff2 /staff2 advfs rw 1 0
```

5. Mount the `staff2` fileset on the `/staff2` directory by entering the following command:

```
# mount -t advfs staff_domain#staff2 /staff2
```

6. Restore the `/staff2` file system from the intermediate file to the `staff2` fileset:

```
# vrestore -xf /tmp/staff_bck -D /staff2
```

The converted `/staff2` file system is ready to use.

### C.3.3 Using a Second Disk

You can convert a data (UFS) file system on one disk to the equivalent data (AdvFS) file system on a different target disk.

The following are required:

- Root user privilege
- A second disk with 5% more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:
  - File system `/staff2`
  - Disk partition `/dev/disk/dsk3g`
- New AdvFS configuration:
  - File system `/staff2`
  - Disk partition `/dev/disk/dsk2c`
  - Mount directory `/staff2`
  - Domain `staff_domain`

Fileset `staff2`

Use the following procedure as a guide for converting the `/staff2` file system:

1. Log in as root on the system containing the `/staff2` file system.
2. Create the `staff_domain` domain and `staff2` fileset:  

```
# mkfdmn /dev/disk/dsk2c staff_domain
# mkfset staff_domain staff2
```
3. Create a mount-point directory and mount the new fileset on the directory:  

```
# mkdir /new_staff2
# mount -t advfs staff_domain#staff2 /new_staff2
```
4. Be certain there is no activity on the system. Copy the contents of the UFS file system to the AdvFS file system:  

```
# vdump -0f - -D /staff2 | vrestore -xf - -D /new_staff2
```
5. Edit the `/etc/fstab` file:
  - a. Search for the entry that mounts `/staff2` as a UFS file system, such as:  

```
/dev/disk/dsk3g /staff2 ufs rw 1 2
```
  - b. Replace it with one that mounts `/staff2` as an AdvFS file system:  

```
staff_domain#staff2 /staff2 advfs rw 1 0
```
6. Unmount `/new_staff2`:  

```
# umount /new_staff2
```
7. Unmount `/staff2`:  

```
# umount /staff2
```
8. Remove the old directory:  

```
# rmdir /staff2
```
9. Mount the new fileset:  

```
# mount /staff2
```

### C.3.4 Using a Second System

You can transfer an existing data file system to a new system, then you can convert the file system to AdvFS.

The following are required:

- Two systems and a common facility for transferring the files such as the `tar` utility (see `tar(1)`)

- Root user privilege on the target system
- Five percent more disk space for the converted file system
- AdvFS installed on the target system

Assumed system configurations are as follows:

- Existing UFS configuration:  
File system `/staff4`
- New AdvFS configuration:  
File system `/staff4`  
Disk partition `/dev/disk/dsk2c`  
Mount directory `/staff4`  
Domain `staff_domain`  
Fileset `staff4`

Use the following procedure as a guide for converting the `staff4` file system:

1. Log in to the system containing the `/staff4` file system and back up the file system to tape:

```
# tar c /staff4
```

2. Log in as root user on the target system.
3. Create the `staff_domain` domain and the `staff4` fileset:

```
# mkfdmn /dev/disk/dsk2c staff_domain
# mkfset staff_domain staff4
```

4. Create a mount-point directory and mount the new fileset on the directory:

```
# mkdir /staff4
# mount -t advfs staff_domain#staff4 /staff4
```

5. Restore the `/staff4` file system from the default tape drive, `/dev/tape/tape0`:

```
# mt rewind
# tar x /staff4
```

6. Edit the `/etc/fstab` file to add an entry that mounts `/staff4` as a UFS file system:

```
staff_domain#staff4 /staff4 advfs rw 1 0
```

The `staff_domain` domain now includes the `staff4` fileset, which is ready to use.

## C.4 Converting from AdvFS to UFS

Converting your entire system from AdvFS to UFS is a multistep process. You first convert the AdvFS root file system to UFS. Then you convert each AdvFS fileset to a UFS file system.

### C.4.1 Converting the root File System to UFS

To convert the root file system, you must mount a UFS disk while your AdvFS root fileset is mounted.

The following are required:

- Root user privilege
- A second bootable disk (You must use partition a.)

Assumed system configurations are as follows:

- Existing AdvFS configuration:  
File system `root`  
Disk partition `/dev/disk/dsk1a`  
Domain `root_domain`  
Fileset `root`
- New UFS configuration:  
File system `root`  
Disk partition `/dev/disk/dsk2a`

Use the following procedure as a guideline for converting your file system:

1. Log in as root user.
2. Create a UFS file system (on a device here called `<disk>`):  

```
# newfs /dev/disk/dsk2a <disk>
```
3. Create a mount-point directory and mount the UFS file system:  

```
# mkdir /newroot  
# mount -t ufs /dev/disk/dsk2a /newroot
```
4. Restore the AdvFS root file system to the `/dev/disk/dsk2a` UFS file system:  

```
# vdump -0f - / | (cd /newroot; vrestore -xf -)
```
5. Make the disk containing the UFS file system a bootable disk:  

```
# disklabel -r /dev/rdisk/dsk2a > /tmp/dsk2label  
# disklabel -t ufs -r -R /dev/rdisk/dsk2a  
/tmp/dsk2label <disk>
```

6. Edit the `/etc/fstab` file on the UFS file system to refer to the new root entry.
  - a. Search `/newroot/etc/fstab` for the entry previously mounted as root for the AdvFS file system:
 

```
root_domain#root / advfs rw 1 0
```
  - b. Replace with one that mounts root as a UFS file system:
 

```
/dev/disk/dsk2a / ufs rw 1 1
```
7. Shut down the system by entering the following command:
 

```
# shutdown -h now
```
8. Reset the boot default device, `BOOTDEF_DEV`, to the new root disk. (Refer to your hardware manual for specific information.)
9. Reboot the system to enable the UFS root file system.

### C.4.2 Converting a Fileset to UFS

Once the root file system is converted to UFS, you can convert your filesets.

The following are required:

- Root user privilege
- A tape or disk for back up

Assumed system configurations are as follows:

- Existing AdvFS configuration:
 

```
Mount directory /staff2
Domain staff_domain
Fileset staff2
```
- New UFS configuration:
 

```
Mount directory /staff2
Disk partition /dev/disk/dsk2c
```

The following example assumes your AdvFS domain contains one volume and only one fileset. If the AdvFS domain contains multiple filesets, then you must create a separate UFS file system for each fileset.



---

### Caution

---

Be sure to perform a full backup on all AdvFS filesets before you start the conversion.

---

1. Make a backup of the AdvFS fileset:

```
# vdump -0f /dev/tape/tape0 /staff2
```

2. Unmount the fileset:

```
# umount /staff2
```

3. Delete the fileset:

```
# rmfset staff_domain staff2
```

4. Remove the domain:

```
# rmfdmn staff_domain
```

5. Create the UFS file system for the specified disk type (on a device here called <disk>). (If the partition you are creating is currently labeled AdvFS, you will get a warning message. Respond **yes** to override the AdvFS designation.)

```
# newfs /dev/disk/dsk2c <disk>
```

6. Edit your `/etc/fstab` file.

- a. Search for the entry that mounts `/staff2` as an AdvFS fileset, such as:

```
staff_domain#staff2 /staff2 advfs rw 1 0
```

- b. Replace it with one that mounts `/staff2` as a UFS file system:

```
/dev/disk/dsk2c /staff2 ufs rw 1 2
```

7. Mount the UFS file system:

```
# mount -t ufs /dev/disk/dsk2c /staff2
```

8. Use the `vrestore` command to load the files from the backup into the UFS file system:

```
# vrestore -xvf /dev/tape/tape0 -D /staff2
```

If your domain contains multiple volumes, you must verify that the disk space allocated to a fileset will not exceed the limit of the UFS file system disk partition. You may need to create multiple UFS file systems to hold the filesets in the domain.



---

## Glossary

This glossary defines some of the terms and acronyms used in the AdvFS documentation.

### **AdvFS Utilities**

A layered product available by license. The utilities allow you to create and manipulate multivolume file domains.

### **advfsd**

The agent that issues commands and obtains system information for the AdvFS GUI.

### **agent**

See **advfsd**. The agent runs in the traditional style of the UNIX daemon.

### **agent state monitor interval**

The time interval between agent scans of the system disk.

### **asynchronous I/O**

A non-blocking I/O scheme where data is written to the cache and may return control before the data is written to the disk.

### **atomic write data logging**

Guarantees that all data in a write system call (up to 8 kilobytes) is either written to the disk or none of the data is written to the disk.

### **balance**

To even the distribution of files between volumes of a domain.

### **bitfile**

A set of pages that AdvFS views as one entity. Reserved files and user files are bitfiles.

### **bitfile metadata table (BMT)**

See **BMT**.

### **block**

A 512-byte unit of disk storage. Sixteen blocks comprise a page.

### **BMT**

Bitfile Metadata Table. An array pages, each with a header and an array of mcells located on each volume. A BMT contains all metadata for all files that have storage on the volume.

**buffer cache**

The area of memory that contains the blocks of data read from and/or waiting to be written to disk.

**checksum**

Blocks created during tape backup for error recovery.

**clone fileset**

A read-only copy of a fileset that preserves the data and structure of an existing fileset at the time the clone is created. Initially the clone uses very little space. The first time data in a block assigned to the original fileset changes, the original block is preserved in the clone. As more disk blocks change, the clone uses more disk space. The contents of the clone fileset can be backed up while the original fileset remains available to users.

**Collect**

A real-time performance monitoring application.

**contiguous**

Storage that is physically adjacent on a disk volume.

**copy-on-write**

The process by which original information is saved in a clone fileset when data in the original file is changed.

**defragment**

To make each file and free space in a domain more contiguous.

**direct I/O**

An I/O scheme that synchronously reads and writes data from a file without copying it to a cache.

**dirty data**

Data that has been written by the application, but the file system has cached it in memory so it has not yet been written to disk.

**domain**

A named pool of storage that contains one or more volumes.

**domain ID**

A set of numbers that identify the domain to the system.

**domain panic**

A condition that prevents further access to the domain when corruption in the domain is detected. AdvFS allows the filesets in the domain to be unmounted after a domain panic.

**DVN**

Domain Version Number. A number in the disk metadata that specifies file structure. Version 5.0 and later domains contain a DVN of 4, while domains created under earlier operating systems have a DVN of 3.

**dtadvfs**

The AdvFS Graphical User Interface (GUI).

**/etc/fdmns directory**

A directory that contains the domain definitions.

**/etc/fstab file**

A file that identifies file systems that are to be mounted at system reboot.

**extent**

Contiguous area of disk space allocated to a file. A file may have zero or more extents.

**extent map**

A table of the size and location of the extents belonging to a file. Simple files have one extent map; striped files have an extent map for every stripe segment.

**file extent**

See **extent**.

**file fragment**

Created when a file uses only part of the last page of file storage allocated or has a total size of less than 8 kilobytes.

**fileset**

A hierarchy of directory and files. A fileset represents a mountable portion of the directory hierarchy of the AdvFS file system.

**fileset quota**

A quota that limits the amount of disk storage that a fileset can consume or the number of files a fileset can contain.

**frag file**

A file that is used to allocate storage for files or file segments that are less than 8 kilobytes (one page). Using fragments reduces the amount of wasted disk space.

**grace period**

The period of time a quota's soft limit can be exceeded as long as the hard limit is not exceeded.

**GUI**

A graphical user interface.

**GUI refresh interval**

The time interval between updates of the GUI window information.

**hard limit**

The quota limit for disk block usage or number of files that cannot be exceeded.

**inode**

A numeric file identifier.

**Logical Storage Manager (LSM)**

Logical Storage Manager is a storage management system that mirrors volumes and provides volume-level striping.

**mcell**

Metadata cells that contain records of file statistics.

**metadata**

File structure information such as file attributes, extent maps, and fileset attributes.

**migrate**

To move files from one volume to another within a domain.

**mirror**

To maintain identical copies of data on different storage areas; one of the copies that is maintained.

**miscellaneous metadata bitfile**

Maps areas of the volume that do not represent AdvFS metadata, such as the disk label and boot blocks.

**NetWorker**

NetWorker for Tru64 UNIX provides scheduled, on-line automated backup.

**object**

A volume, domain, fileset, or clone fileset managed by the AdvFS GUI.

**object tree**

The AdvFS GUI hierarchical display of objects.

**page**

An allocation of 8 kilobytes of contiguous disk space (16 blocks).

**product authorization key (PAK)**

License to access Compaq Computer Corporation software.

**quota file**

A file that stores quota limits and keeps track of number of files, disk block usage, and grace period per user ID or per group ID. Fileset quota information is stored within the fileset metadata.

**root tag file**

A bitfile that defines the location of all filesets in a domain. Each domain has one.

**saveset**

A collection of blocks created by the `vdump` utility to save AdvFS backup information.

**segment**

See **stripe segment**.

**soft limit**

The quota value beyond which disk block usage or number of files is allowed only during the grace period.

**sparse file**

A file whose pages do not all have allocated disk space.

**storage area**

See volume.

**storage bitmap**

Keeps track of allocated disk space on a volume.

**stripe**

To distribute data across multiple storage areas. AdvFS stripes individual files. LSM/RAID stripes storage across multiple devices.

**stripe segment**

For AdvFS, the portion of a striped file that resides on a volume. A file striped across four volumes has four stripe segments. Segments can be migrated from one volume to another.

**tag**

A unique identifier for an AdvFS file within a fileset.

**transaction log**

The log that records changes to metadata before the changes are written to disk. At regular intervals these changes are written to disk.

**trashcan**

A directory that contains the most recently deleted files from an attached directory. Trashcan directories can be set up by each user for user files.

**Unified Buffer Cache (UBC)**

The dynamically allocated system buffer cache that holds file data and AdvFS metadata.

**volume**

For AdvFS, anything that behaves like a UNIX block device. This can be a disk, disk partition, or logical volume.

**write-ahead logging**

The process by which the modifications to the file-structure information are completely written to a transaction log before the actual changes are written to disk.



---

# Index

## A

---

- adding volume, 1-13, 5-4, 6-18
- addvol command, 1-13
- AdvFS Utilities commands, B-3
- AdvfsAccessMaxPercent attribute, 4-20
- advfsd command, 6-2
- AdvfsDomainPanicLevel attribute, 4-21
- AdvfsReadyQLim attribute, 4-20
- advfsstat command, 4-7
- AdvfsSyncMmapPages attribute, 4-20
- advscan command, 5-17
- agent, 6-2
- asynchronous I/O, 4-4
- atomic-write data logging, 4-5, 5-2
- attributes, 4-20

## B

---

- backup
  - cloning, 3-6
  - data, 3-2
  - database, 3-6
  - error protection, 3-4
  - incremental, 3-5
  - remote, 3-5
- balance command, 4-12
- balancing domains, 4-12, 5-3, 6-19
- bitfile metadata table
  - ( See BMT )
- bitfile, miscellaneous metadata, 1-6
- bitmap, storage, 1-6
- BMT, 1-6

## C

---

- cache tuning, 4-20
- chfile command, 4-4, 5-2
- chfsets command, 2-17, 2-23
- chvol command, 1-7, 4-20
- clear bit error message, 5-11
- clone
  - creating, 1-25, 6-16
  - deleting, 1-26, 6-17
  - in backup, 3-6
  - mounting, 1-26, 6-16
  - removing, 1-26, 6-17
  - renaming, 1-26
  - unmounting, 1-26, 6-17
- clonefs command, 1-25
- cloning, 1-24
- Collect, 4-7
- commands
  - addvol, 1-13
  - AdvFS Utilities, B-3
  - advfsd, 6-2
  - advfsstat, 4-7
  - advscan, 5-17
  - balance, 4-12
  - base system, B-1
  - chfile, 4-4, 5-2
  - chfsets, 2-17, 2-23
  - chvol, 1-7, 4-20
  - clonefs, 1-25
  - defragment, 4-7
  - df, 5-4
  - dtadvfs, 6-3
  - du, 5-4
  - edquota, 2-4, 2-9, 2-18
  - iostat, 4-7

- migrate, 4-15
- mkfdmn, 1-11
- mkfset, 1-21
- mktrashcan, 4-21
- mount, 1-22
- mountlist, B-2
- ncheck, 2-11
- quot, 2-11
- quota, 2-12
- quotacheck, 2-13, 5-5
- renamefset, 1-23
- repquota, 2-13
- rmfdmn, 1-17
- rmfset, 1-22, 1-26
- rmtrashcan, 4-21
- rmvol, 1-15
- rvdump, 3-1
- rvrestore, 3-1
- salvage, 5-8
- savemeta, 5-25
- shfragbf, B-2
- showfdmn, 1-11, 5-4
- showfile, 4-16
- showfsets, 1-20, 2-22
- shtrashcan, 4-21
- stripe, 4-17
- switchlog, 4-2
- tag2name, B-2
- UFS comparison, B-4
- umount, 1-22
- vdf, 2-21
- vdump, 3-1
- verify, 5-6
- vrestore, 3-1
- compressing savesets, 3-4
- configuring
  - AdvFS root, 1-26
  - domain, 1-7, 4-1
  - file system, 1-5
  - fileset, 1-18
- converting
  - AdvFS to UFS
    - fileset, C-14

- root, C-13
  - UFS to AdvFS
    - data, C-7
    - root, C-6
    - /usr, C-1
- copy-on-write, 1-24
- corruption
  - due to old utilities, 5-1
- crash recovery, 5-25
- creating
  - clone, 1-25, 6-16
  - domain, 1-11, 6-9
  - fileset, 1-21, 6-13

## D

---

- data
  - backup, 3-2
  - converting UFS to AdvFS, C-7
  - logging, 4-4, 5-2
  - restoring, 3-7
- database backup, 3-6
- defragment command, 4-7
- defragmenting, 4-7, 5-3, 6-19
- deleting
  - clone, 1-26, 6-17
  - domain, 1-17, 6-12
  - fileset, 1-22, 6-15
  - fileset quotas, 2-23
  - user and group quotas, 2-16
  - volume, 1-15, 6-18
- Devices and Volumes window, 6-8
- df command, 5-4
- direct I/O, 4-6, 5-2
- directory, root tag, 1-7
- disk
  - access incompatibility, 5-3
  - bad usage information, 5-5
  - checking usage, 2-10
  - damaged, 5-26
  - domain panic, 5-12
  - domains, 5-1
  - failure recovery, 5-20
  - moving, 5-26

- partitions, dumping to, 3-3
- structure, 5-25
- disk structure
  - examining, 5-25
- disks.ignore file, 6-4
- domain, 1-7
  - adding volumes, 5-4, 6-11
  - balancing, 4-12, 5-3, 6-19
  - changing name of root, 1-27
  - configuring, 1-7, 4-1
  - creating, 1-11, 6-9
  - damaged, 5-8
  - defragmenting, 4-7, 5-3, 6-19
  - deleting, 1-17, 6-12
  - description, 1-2
  - displaying information, 1-11
  - enlarging, 1-13, 6-11
  - exchanging volumes, 5-4
  - migrating, 5-3
  - multivolume setup, 1-13
  - panic, 5-12
  - recovering, 5-8
    - multivolume domain, 5-23
    - root domain, 5-22
  - removing, 1-17, 6-12
  - removing volumes, 5-4
  - renaming, 1-17, 6-12
  - repairing, 5-17
  - root, 1-26
  - security, 1-17, 6-12
  - setting free space alert, 6-11
  - shrinking, 1-15
  - single volume set up, 1-11
  - space usage, 5-4
  - striping, 4-17
  - upgrading, 1-9, 5-3
  - Version 5 changes, 1-9
  - version incompatibility, 5-1
- domain panic, 5-12
- domain version number, 1-9
- dtadvfs command, 6-3

- du command, 5-4
- dumping
  - error protection, 3-4
  - incremental, 3-5
  - remote, 3-5
  - subdirectories, 3-3
  - to disk partitions, 3-3
  - to tape, 3-3
- DVN, 1-9

## E

---

- edquota command, 2-4, 2-9, 2-18
- enlarging a domain, 1-13, 6-11
- /etc/fdmns directory, 1-8, 1-27
  - restoring, 5-15
- /etc/fstab, 1-19
  - changing fileset name, 1-27
  - quotas, 2-4, 2-15
- extent, 1-3

## F

---

- file
  - deleting
    - trashcan, 4-21
  - disks.ignore, 6-4
  - extent, 1-3
  - fragments, 1-4
  - GUI log, 6-4
  - gui.passwd, 6-3
  - hosts.allow, 6-3
  - moving, 4-14
  - preallocation, 1-4
  - quotas, 2-3, 2-17
  - restoring, 3-8
  - sparse, 1-4
  - storage allocation, 1-3
  - striping, 4-17, 5-3
  - structure, new, 1-9, 5-1
- file quotas

- ( See quotas )
- file system
  - AdvFS root, 1-26
  - configuring, 1-1
  - converting AdvFS to UFS, C-13
  - converting UFS to AdvFS
    - data, C-7
    - root, C-6
    - /usr, C-1
  - design, 1-1
  - improving performance, 5-2
  - monitoring remotely with GUI, 6-5
  - recovering, 5-25
  - tuning, 4-7
  - verifying, 5-6
  - viewing with GUI, 6-6
- fileset, 1-18
  - changing name of root, 1-27
  - cloning, 1-24
  - configuring, 1-18
  - converting AdvFS to UFS, C-14
  - creating, 1-21, 6-13
  - deleting, 1-22, 6-15
  - description, 1-2
  - displaying information, 1-20
  - mounted read-only, 5-14
  - mounting, 1-22, 5-27, 6-14
  - quotas, 6-15
  - read-only recovery, 5-14
  - removing, 1-22, 6-15
  - renaming, 1-23, 6-15
  - security, 1-22, 6-15
  - setting free space alert, 6-14
  - space usage, 5-4
  - unmounting, 1-22, 6-14
  - upgrading, 1-21
- fragments, 1-4
- free space alert, 6-7
  - domain, 6-11
  - fileset, 6-14
  - volume, 6-18

## G

---

- grace period, 2-2
  - setting fileset, 2-18
  - setting user and group, 2-4
  - turning off, 2-17
- graphical user interface
  - ( See GUI )
- group quotas, 2-3
- growing a domain, 6-11
- GUI
  - agent, 6-2
  - allowed hosts, 6-3
  - and LSM volumes, 6-5
  - clone
    - creating, 6-16
    - deleting, 6-17
    - mounting, 6-16
    - removing, 6-17
    - unmounting, 6-17
  - Devices and Volumes window, 6-8
  - domain
    - adding volumes, 6-11
    - balancing, 6-19
    - creating, 6-9
    - defragmenting, 6-19
    - deleting, 6-12
    - enlarging, 6-11
    - free space alert, 6-11
    - removing, 6-12
    - renaming, 6-12
  - fileset
    - creating, 6-13
    - deleting, 6-15
    - free space alert, 6-14
    - mounting, 6-14
    - removing, 6-15
    - renaming, 6-15
    - setting quotas, 6-15
    - unmounting, 6-14
  - ignoring disks, 6-4
  - installing, 6-1
  - log files, 6-4
  - Main window, 6-7

- monitoring remote file systems, 6-5
- password file, 6-3
- security, 6-3
- software subsets, 6-1
- troubleshooting, 6-19
- volume
  - adding, 6-18
  - removing, 6-18
  - setting free space alert, 6-18
- GUI refresh interval, 6-3
- gui.passwd file, 6-3

## H

---

- hosts.allow file, 6-3

## I

---

- I/O
  - asynchronous, 4-4
  - atomic-write, 4-5
  - direct, 4-6
  - synchronous, 4-5
- installing
  - AdvFS GUI, 6-1
  - Version 5 upgrade, 1-9, 5-1
- iostat command, 4-7

## L

---

- license
  - restoring, 5-23
- limits, quota, 2-1
- log, transaction
  - ( See transaction log )
- Logical Storage Manager
  - ( See LSM )
- LSM
  - and AdvFS, 1-29
  - mirror, 1-31
  - striping, 1-32
  - and GUI, 6-5

## M

---

- memory mapping, 5-2
- metadata, 1-6
- metadata, displaying, 5-25
- migrate command, 4-15
- migrating, 4-15, 5-3
- mirror, 1-31
- miscellaneous metadata bitfile, 1-6
- mkfdmn command, 1-11
- mkfset command, 1-21
- mktrashcan command, 4-21
- mount command, 1-22
- mounting
  - clone, 1-26, 6-16
  - fileset, 1-22, 6-14
- mountlist command, B-2
- moving
  - between AdvFS and UFS, C-1
  - disk, 5-26

## N

---

- naming conventions, 1-2
- ncheck command, 2-11
- NetWorker, 3-9

## P

---

- panic, domain, 5-12
- partition
  - finding AdvFS, 5-17
  - overlapping, 1-15
  - reusing, 5-5
- performance
  - improving, 5-2
  - monitoring, 4-7

## Q

---

- quot command, 2-11
- quota

- running into limits, 5-6
- quota command, 2-12
- quotacheck command, 2-13, 5-5
- quotas, 2-1
  - activating fileset, 2-23
  - activating user and group
    - at start-up, 2-15
    - manually, 2-15
  - changing GUI view, 6-7
  - checking disk usage, 2-10
  - checking disk usage for filesets, 2-20
  - checking fileset ownership, 2-11
  - deactivating fileset, 2-23
  - deactivating user and group, 2-16
  - deleting fileset, 2-23
  - deleting user and group, 2-16
  - disabling fileset, 2-23
  - disabling user and group, 2-16
  - displaying disk usage, 2-12
  - enabling fileset, 2-23
  - enabling user and group
    - at start-up, 2-15
    - manually, 2-15
  - /etc/fstab, 2-4, 2-15
  - file, 2-3, 2-17
  - fileset, 2-17, 6-7, 6-15
  - grace period, 2-2
  - group, 2-3
  - hard limits, 2-1
  - relocating files, 2-4
  - removing fileset, 2-23
  - removing user and group, 2-16
  - restoring, 3-8
  - setting fileset, 2-17, 6-15
  - setting for multiple filesets, 2-19
  - setting for multiple users and groups, 2-9
  - setting user and group, 2-4
  - soft limits, 2-1
  - starting fileset, 2-23
  - starting user and group
    - at start-up, 2-15

- manually, 2-15
- summarizing by fileset, 2-13
- turning off fileset, 2-23
- turning off user and group, 2-16
- users, 2-3
- viewing with GUI, 6-7

## R

---

- read-only fileset, 5-14
- recovering
  - domain, 5-8
  - fileset mounted read-only, 5-14
  - multivolume domain, 5-23
  - root domain, 5-22
  - system crash, 5-25
- removing
  - clone, 1-26, 6-17
  - domain, 1-17, 6-12
  - fileset, 1-22, 6-15
  - fileset quotas, 2-23
  - striping, 4-19
  - user and group quotas, 2-16
  - volume, 1-15, 6-18
- renamefset command, 1-23
- renaming
  - clone, 1-26
  - domain, 1-17, 6-12
  - fileset, 1-23, 6-15
- repairing
  - domain, 5-17
- repquota command, 2-13
- restoring
  - data, 3-7
  - /etc/fdmns directory, 5-15
  - files, 3-8
  - license, 5-23
  - quotas, 3-8
  - saveset, 3-8
- rmfdmn command, 1-17
- rmfset command, 1-22, 1-26
- rmtrashcan command, 4-21
- rmvol command, 1-15
- root

- changing domain name, 1-27
- changing fileset name, 1-27
- converting AdvFS to UFS, C-13
- converting UFS to AdvFS, C-6
- domain, 1-26
- recovering, 5-22
- root tag directory, 1-7
- rvdump command, 3-1
- rvrestore command, 3-1

## S

---

- salvage command, 5-8
- savemeta, 5-25
- saveset, 3-2
  - compressing, 3-4
  - restoring, 3-8
- security
  - domain, 1-17, 6-12
  - fileset, 1-22, 6-15
- setting up
  - AdvFS, 1-5
  - multivolume domain, 1-13
  - single-volume domain, 1-11
- shfragbf, B-2
- showfdmn command, 1-11, 5-4
- showfile command, 4-16
- showfsets command, 1-20, 2-22
- shrinking a domain, 1-15
- shtrashcan command, 4-21
- smooth sync, 4-20
- space
  - corrupt disk usage output, 5-5
  - domain, 5-4
  - fileset, 5-4
- sparse file, 1-4
- storage bitmap, 1-6
- stripe command, 4-17
- striping, 4-17
  - choosing AdvFS or LSM, 1-32

- file, 4-17, 5-3
- removing, 4-19
- switchlog command, 4-2
- synchronous I/O, 4-5
- SysMan Menu, A-1

## T

---

- tag2name command, B-2
- transaction log, 1-7
  - data logging, 4-4
  - description, 1-3
  - improving performance, 4-2
  - incompatibility, 5-2, 5-27
  - isolating, 4-3
- trashcan, 4-21
- trashcan commands, 4-21
- tuning
  - data cache, 4-20
  - file system, 4-7

## U

---

- UBC, 4-20
- UFS
  - AdvFS command comparison, B-4
  - converting AdvFS to, C-13
  - converting fileset to, C-14
- umount command, 1-22
- units, GUI display, 6-7
- unmounting
  - clone, 1-26, 6-17
  - fileset, 1-22, 6-14
- upgrading
  - domain, 1-9
  - fileset, 1-21
- user quotas, 2-3
- /usr
  - converting UFS to AdvFS, C-1
- utility incompatibility, 5-1

## V

---

- vdf command, 2-21
- vdump command, 3-1
- verify command, 5-6
- verifying, file system, 5-6
- version number, domain, 1-9
- volume, 1-6
  - adding, 1-13, 5-4, 6-18
  - attributes, 1-7
  - changing, 4-14
  - deleting, 1-15, 6-18
  - description, 1-2
  - exchanging, 5-4
  - removing, 1-15, 5-4, 6-18
  - reusing, 5-5
  - setting free space alert, 6-18
  - structure, 1-6
- vrestore command, 3-1



---

## How to Order Tru64 UNIX Documentation

To order Tru64 UNIX documentation in the United States and Canada, call **800-344-4825**. In other countries, contact your local Compaq subsidiary.

If you have access to Compaq's intranet, you can place an order at the following Web site:

<http://asmorder.nqo.dec.com/>

If you need help deciding which documentation best meets your needs, see the Tru64 UNIX Documentation Overview, which describes the structure and organization of the Tru64 UNIX documentation and provides brief overviews of each document.

The following table provides the order numbers for the Tru64 UNIX operating system documentation kits. For additional information about ordering this and related documentation, see the Documentation Overview or contact Compaq.

Name	Order Number
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
Startup Documentation Kit	QA-6ADAC-GZ
General User Documentation Kit	QA-6ADAD-GZ
System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAF-GZ
Reference Pages Documentation Kit	QA-6ADAG-GZ
TruCluster Server Documentation Kit	QA-6BRAA-GZ



---

## Reader's Comments

### Tru64 UNIX

AdvFS Administration

AA-RH96B-TE

Compaq welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

- This postage-paid form
- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: (603) 884-0120, Attn: UBPG Publications, ZK O3-3/Y32

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

#### Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usability (ability to access information quickly)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Please list errors you have found in this manual:

Page	Description
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

#### Additional comments or suggestions to improve this manual:

---

---

---

---

What version of the software described by this manual are you using? 

---

Name, title, department 

---

Mailing address 

---

Electronic mail 

---

Telephone 

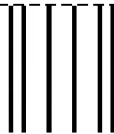
---

Date 

---

----- Do Not Cut or Tear - Fold Here and Tape -----

**COMPAQ**



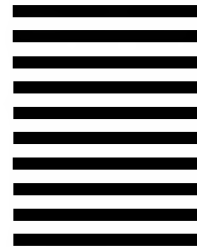
NO POSTAGE  
NECESSARY IF  
MAILED IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ COMPUTER CORPORATION  
UBPG PUBLICATIONS MANAGER  
ZKO3-3/Y32  
110 SPIT BROOK RD  
NASHUA NH 03062-2698



----- Do Not Cut or Tear - Fold Here -----

Cut on This Line